



**UNIVERSIDAD NACIONAL DE INGENIERÍA  
RECINTO UNIVERSITARIO SIMÓN BOLÍVAR  
FACULTAD DE ELECTROTECNIA Y COMPUTACIÓN**



**TRABAJO MONOGRAFICO PARA OPTAR AL TITULO DE INGENIERIO  
EN ELECTRONICA**

**TÍTULO:**

**Implementación de un Sistema de Posicionamiento Vehicular utilizando  
tecnologías GPS y GLONASS.**

**AUTORES:**

Br. Marcos Daniel Mercado Jirón.

Br. Erving Javier Gutiérrez Moreno.

**TUTOR:**

TeknL Marco A. Munguía Mena

Agosto, 2016



## **Agradecimientos.**

Agradezco a mis padres; Mi madre **Marcia A. Jirón Cortez** y a mi padre **Marcos A. Mercado García** por el apoyo tanto económico como emocional, es gracias a ellos que hoy puedo presentar este trabajo para optar al título de ingeniero, igualmente agradezco a mi hermana menor **Kenia M. Mercado Jirón** la cual ha sido un apoyo muy importante.

También agradezco a nuestro tutor **Marco Munguía Mena** por escuchar nuestras ideas, quien mediante críticas constructivas nos ayudó enormemente con este trabajo y por último pero no menos importante agradezco a mi país la República de Nicaragua.

*Marcos D. Mercado Jirón*

## **Agradecimientos.**

Primero y antes que nada, dar gracias a Dios, por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

También agradecer a mi familia que fueron fuente de apoyo constante e incondicional en toda mi vida y más aún en mis duros años de carrera profesional y en especial quiero expresar mi más grande agradecimiento a mi madre **Lloyda J. Moreno G.** que sin su ayuda hubiera sido imposible culminar mi profesión.

Por último agradecer a mi tutor al Ing. **Marco Munguía** por haberme brindado la oportunidad de recurrir a su capacidad y conocimiento científico, así como también haberme tenido toda la paciencia del mundo para guiarme durante todo el desarrollo de la monografía.

*Erving J. Gutiérrez Moreno*



## Resumen

En el presente documento se presenta el diseño e implementación de un dispositivo AVL utilizando GPS/GLONASS y GPRS. El dispositivo AVL está compuesto por un microcontrolador PIC24/33, módulo GSM/GPRS SIM900, módulo GNSS (GPS/GLONASS) L76-M33/L16, una tarjeta SD para el almacenamiento de los datos y una página web que utiliza PHP, MySQL, JavaScript/AJAX y Google Maps.

El microcontrolador se encarga de controlar el módulo GSM/GPRS mediante comandos AT y controlar también el módulo GNSS mediante el protocolo NMEA 0183 y PMTK. Además, el microcontrolador es responsable de respaldar la información en una tarjeta SD. Una vez la información es procesada localmente se envía al servidor, este la formatea y la almacena en una base de datos MySQL para su posterior visualización por los usuarios en un navegador web.

Entre los resultados obtenidos se destacan varias pruebas realizadas exitosamente en donde el sistema opera correctamente en ambientes semi-urbanos, urbanos y de carretera cuyo trayecto incluye partes urbanas y partes rurales de densa vegetación.

## Acrónimos

<b>AJAX</b>	Asynchronous JavaScript And XML
<b>ANSI</b>	American National Standards Institute
<b>ARM</b>	Advanced RISC Machine
<b>AT</b>	Attention (Comandos)
<b>AUC</b>	Authentication Centre
<b>AVL</b>	Automatic vehicle location
<b>BSC</b>	Base station controller
<b>BSS</b>	base station subsystem
<b>BTS</b>	Base transceiver station
<b>C/A</b>	Coarse/Acquisition
<b>CDMA</b>	Code Division Multiple Access
<b>CRC</b>	Cyclic Redundancy Check
<b>CSS</b>	Cascading Style sheets
<b>dB</b>	Decibel
<b>DGPS</b>	Differential GPS
<b>DOM</b>	Document Object Model
<b>DTMF</b>	Dual-Tone Multi-Frequency
<b>EGNOS</b>	European Geostationary Navigation Overlay System
<b>EIR</b>	Equipment Identity Register
<b>EMI</b>	Electromagnetic Interference
<b>ESA</b>	European Space Agency
<b>FAT</b>	File Allocation Table
<b>FDMA</b>	Frequency Division Multiple Access
<b>FEC</b>	Forward Error Correction
<b>FTP</b>	File Transfer Protocol
<b>GBAS</b>	Ground Based Augmentation System
<b>GCC</b>	Galileo Control Center
<b>GCS</b>	Ground Control Segment
<b>GGSN</b>	Gateway GPRS support node
<b>GLONASS</b>	Global Navigation Satellite System
<b>GMS</b>	Galileo Mission Segment
<b>GNSS</b>	Global Navigation Satellite System
<b>GPRS</b>	General Packet Radio Service
<b>GPS</b>	Global Positioning System
<b>GSM</b>	Global System for Mobile communications
<b>HLR</b>	Home location register
<b>HTML</b>	HyperText Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>I2C</b>	Inter-Integrated Circuit
<b>ICSP</b>	In-Circuit Serial Programming
<b>IP</b>	Internet Protocol
<b>ISDN</b>	Integrated Services Digital Network
<b>ISO</b>	International Organization for Standardization
<b>ITU</b>	International Telecommunication Union
<b>JS</b>	JavaScript
<b>LAAS</b>	Local Area Augmentation System

<b>LAMP</b>	Linux Apache MySQL PHP
<b>LNA</b>	Low-noise amplifier
<b>MCS</b>	Master Control Station
<b>MFSAS</b>	Multi-functional Satellite Augmentation System
<b>MIPS</b>	millions of instructions per second
<b>MMC</b>	Multi Media Card
<b>MS</b>	Mobile station
<b>MSB</b>	Most Significant Bit
<b>MSC</b>	Mobile switching center
<b>NASA</b>	National Aeronautics and Space Administration
<b>NCF</b>	Navigation Control Facility
<b>NMEA</b>	National Marine Electronics Association
<b>NMS</b>	Network Management System
<b>NSCC</b>	Navigation System Control Centre
<b>NSS</b>	Network SubSystem
<b>OEM</b>	Original Equipment Manufacturer
<b>PCB</b>	Printed Circuit Board
<b>PDN</b>	Packet Data Network
<b>PHP</b>	Hypertext Preprocessor
<b>PPP</b>	Point-to-Point Protocol
<b>PPS</b>	Precise Positioning Service
<b>PSTN</b>	public switch transmission network
<b>RF</b>	Radio Frequency
<b>RFI</b>	Radio Frequency Interference
<b>ROSCOSMOS</b>	Agencia Espacial Federal Rusa
<b>RNSS</b>	Radio Navigation Satellite System
<b>SA</b>	Selective Availability
<b>SBAS</b>	Satellite Based Augmentation System
<b>SCF</b>	Navigation Control Facility
<b>SGSN</b>	Serving GPRS Support Node
<b>SDMA</b>	Space Division Multiple Access
<b>SDC</b>	Secure Digital Card
<b>SMA</b>	SubMiniature version A
<b>SMD</b>	Surface Mount Device
<b>SMS</b>	Short Message Service
<b>SNR</b>	Signal to Noise Ratio
<b>SPI</b>	Serial Peripheral Interface
<b>SQL</b>	Structured Query Language
<b>SSL</b>	Secure Sockets Layer
<b>TCP</b>	Transmission Control Protocol
<b>TTL</b>	transistor-transistor logic
<b>UART</b>	Universal Asynchronous Receiver-Transmitter
<b>UDP</b>	User Datagram Protocol
<b>UMTS</b>	Universal Mobile Telecommunications System
<b>UTC</b>	Universal Time Coordinated
<b>VLR</b>	Visitor Location registers
<b>VPS</b>	virtual private server
<b>WASS</b>	Wide Area Augmentation System
<b>XML</b>	eXtensible Markup Language.

## Lista de Figuras

Figura 1.1	Bloques que integran el dispositivo AVL.....	16
Figura 1.2	Satélites GPS + GLONASS en órbita.....	17
Figura 1.3	Vista de satélites GPS y GLONASS.....	18
Figura 1.4	Constelación de satélites GPS.....	19
Figura 1.5	Sistema GNSS con receptores GPS.....	20
Figura 1.6	Constelación de satélites GLONASS.....	23
Figura 1.7	Red GSM.....	24
Figura 1.8	Conexión GPRS.....	26
Figura 1.9	Conexión ICSP – PICKIT3.....	29
Figura 1.10	SD/ MicroSD pines.....	30
Figura 1.11	Conexión al sitio WEB.....	31
Figura 2.1	Módulo MIKROE-1887 con L76-M33 Quectel.....	38
Figura 2.2	Circuito MIKROE-1887 con L76-M33 Quectel.....	40
Figura 2.3	Conexión de la interfaz serial L16/L76.....	40
Figura 2.4	Antena activa AA161.301111.....	41
Figura 2.5	Diagrama interno, antena activa AA161.301111.....	41
Figura 2.6	Módulo SIM900 Quad-Band.....	43
Figura 2.7	Conexión de la interfaz serial SIM900.....	44
Figura 2.8	Antena GSM QuadBand.....	44
Figura 2.9	Diagrama de estados para conexión GPRS única.....	49
Figura 2.10	Microcontrolador PIC24FJxxGA002.....	51
Figura 2.11	Circuito básico interfaz SPI – SDC.....	51
Figura 2.12	Modelo de Capas para FatFs.....	52
Figura 2.13	Manejo de sistema Único/Múltiple.....	53
Figura 2.14	Diseño página principal.....	55
Figura 2.15	Diseño validación usuario.....	56
Figura 2.16	Diseño vista tracking.....	56
Figura 2.17	Bloques del dispositivo AVL.....	59
Figura 2.18	Conexiones, descripción general.....	59
Figura 2.19	Dependencia de archivos en las Capas FatFs.....	62
Figura 2.20	Sistema AVL (Descripción general).....	65
Figura 2.21	GET y respuesta al servidor.....	67
Figura 2.22	Diagrama en bloques, operación del dispositivo AVL.....	68
Figura 2.23	Circuito dispositivo AVL.....	69
Figura 3.1	Tracking de moto en la ciudad de Masatepe.....	70
Figura 3.2	Tracking de automóvil en la ciudad de Managua.....	71
Figura 3.3	Tracking de microbús ruta Masatepe-Managua.....	72

## Lista de Tablas

Tabla 1.1	Código respuesta HTTP .....	36
Tabla 2.1	Lista de módulos GNSS evaluados.....	37
Tabla 2.2	Lista de módulos GSM/GPRS evaluados .....	42
Tabla 2.3	Comandos AT de inicio .....	45
Tabla 2.4	Comandos AT inicio GPRS.....	45
Tabla 2.5	Comandos AT de conexión GPRS .....	46
Tabla 2.6	Comandos AT conexión sitio web .....	46
Tabla 2.7	Comandos AT Envío/Cierre/Desconexión .....	47
Tabla 2.8	Otros comandos AT .....	48
Tabla 2.9	Lista de microcontroladores evaluados .....	50
Tabla 2.10	Principales funciones de la librería FatFs .....	54
Tabla 2.11	GET/POST HTTP.....	66
Tabla 2.12	Campos enviados al servidor.....	67

## Índice

I. Introducción .....	12
II. Antecedentes.....	13
III. Justificación .....	14
IV. Objetivo General .....	15
VI. Objetivos Específicos.....	15
Capítulo 1 Tecnologías del Sistema AVL .....	16
1.1.Sistema AVL .....	16
1.2 GNSS.....	17
1.3.GPS.....	18
1.4.Protocolo NMEA .....	21
1.5.GLONASS.....	23
1.6.GSM/GPRS .....	24
1.7.Comandos AT .....	27
1.8.Microcontrolador.....	27
1.9.Secure Digital SDC.....	30
1.10.Interfaz de usuario.....	31
1.11.Tecnologías Sitio Web.....	32
1.11.1.Cookies .....	32
1.11.2.HTML/CSS .....	32
1.11.3.PHP .....	32
1.11.4.JavaScript/AJAX .....	33
1.11.5.Google Maps API .....	34
1.11.6.Base de Datos .....	34
1.11.7.HTTP.....	35

Capítulo 2 Diseño del Sistema .....	37
2.1.Subsistema GNSS .....	37
2.2 Subsistema GSM/GPRS .....	41
2.3. Subsistema Microcontrolador .....	49
2.4. Subsistema Almacenamiento .....	51
2.5 Subsistema Visualización .....	55
2.6.Integración .....	59
2.7.Funcionamiento general.....	65
Capítulo 3 Resultados .....	70
Capítulo 4 Conclusiones y Recomendaciones .....	73
4.1.Conclusiones .....	73
4.2.Recomendaciones.....	74
VI. Bibliografía .....	75
VII. Anexos .....	78

## I. Introducción

Los sistemas AVL (Automatic Vehicle Location por sus siglas en Ingles) son usados para la gestión y monitoreo de flotas vehiculares, como una herramienta de seguridad en caso de robo de la unidad y otras aplicaciones más novedosas como la notificación temprana de accidentes. Para este fin, al sistema AVL se le incluyen sensores que determinan si la unidad vehicular sufrió una colisión, en dicho caso el sistema automáticamente notifica a las estaciones médicas más cercanas al sitio para socorrer al conductor.

Un Sistema de rastreo vehicular comúnmente está compuesto por una red de satélites que brindan el servicio de geo-localización, una red de telefonía celular que brinde el servicio GSM(Global System for Mobile communications por su siglas en inglés)/GPRS(General Packet Radio Service por sus siglas en ingles), además los dispositivos de usuario que se encargan de recibir los datos GNSS(Global Navigation Satellite System por sus siglas en ingles) y enviarlos por la red celular mediante GPRS, y por ultimo una interfaz (por lo general web) en donde el usuario pueda ver la información sobre el vehículo rastreado.

La principal constelación de satélite que utilizan los sistemas AVL disponibles en el mercado es la denominada GPS (Global Positioning System, por sus siglas en inglés). Sin embargo esto ya está cambiando con la llegada de nuevos sistemas GNSS como lo son GLONASS, GALILEO, BeiDOu, etc. En la actualidad cada vez es más común encontrar módulos que sean capaces de conectarse simultáneamente a dos o más constelaciones. [1]

En el presente trabajo, implementamos un sistema AVL que utiliza GPS y GLONASS. El GPS es el sistema norteamericano que incluye 24 satélites en 6 planos orbitales con señales CDMA (Code Division Multiple Access, por sus siglas en ingles) y el GLONASS es el sistema Ruso que incluye 24 satélites en 3 planos orbitales con señales FDMA (Frequency Division Multiple Access, por sus siglas en ingles).[2]



## II. Antecedentes

A la fecha se han desarrollado varios prototipos (Open Hardware/Source) a nivel de integración de módulos. No obstante, en el país no hay antecedentes de diseño de este tipo de sistemas. En el extranjero los sistemas opensource más destacables que se comercializan en la actualidad son los desarrollados por MCELECTRONICS y TIGAL.

El TrackMe (GPS) por MCELECTRONICS, es una placa PCB unificada que incluye módulo GPS, SIM900 y PIC18. Las coordenadas son enviadas a través de mensajes de texto (SMS, por sus siglas en inglés) y visualizadas en un SmartPhone con ayuda de Google Maps. El módulo GPS está basado en el SIRFStar III el cual posee baja inmunidad en escenarios con múltiples fuentes de RF. Dicho proyecto contó con la ayuda de MICROCHIP [3]

En el caso del OpenTracker de TIGAL (República Checa), este es un proyecto comercializable completo, usa receptor GPS/GLONASS y microcontrolador AVR. Además es compatible con Arduino y funciona con una interface web para rastreo usando GoogleMaps u OpenStreetMaps. Este proyecto usa receptores GSM/GPRS y GNSS de Quectel. Una de sus limitantes es que para poder conectar un dispositivo AVL a la interfaz WEB, dicho dispositivo debió haber sido manufacturado por TIGAL en caso contrario no se puede conectar y se necesitaría otra interfaz WEB. En su primera versión este dispositivo usaba respaldo de datos mediante tarjeta SD. Sin embargo, en la posterior versión carece de respaldo local. El OpenTracker usa GPRS para conectarse al servidor haciendo GET por el puerto 80 (HTTP) enviando datos cada minuto [4].

### **III. Justificación**

En el país las empresas utilizan dispositivos AVL diseñados en el extranjero que únicamente usan tecnología GPS, desaprovechando así la precisión, confiabilidad y robustez que brindan dispositivos multi-sistemas GNSS.

Mientras que la constelación de GPS ofrece entre 21 y 24 satélites en funcionamiento. Esta cantidad se duplica a los 48 si sumamos los del sistema GLONASS. Esto permite tener en escenarios difíciles de alta urbanización o densa vegetación, entre 10 y 12 satélites en vista. Duplicando el número de satélites si solamente se contase con el sistema GPS. En estas situaciones el potencial de GLONASS mejora la capacidad del GPS proporcionándole más observaciones y mejorando el rendimiento conjunto del sistema. Un receptor GPS +GLONASS trata las dos constelaciones como una sola. Así que la importancia radica en mejor tiempo de adquisición, confiabilidad de lectura, mejor inmunidad en terrenos bastante urbanizados. [5]

Así mismo, Nicaragua actualmente tiene un acuerdo con Rusia para la explotación conjunta de GLONASS, en dicho acuerdo se establece la instalación de una estación de corrección para el sistema de navegación GLONASS en nuestro país lo que mejoraría considerablemente la precisión en Centro América y el Caribe. [6]

#### **IV. Objetivo general**

- Implementar un sistema de posicionamiento que integre las tecnologías GPS/GLONASS para ubicación, seguimiento y monitoreo de vehículos.

#### **V. Objetivos específicos**

- Estudiar y comprender las distintas tecnologías que convergen en el desarrollo de un sistema de rastreo vehicular automático.
- Diseñar e implementar una interfaz de control transmisión-recepción por medio de un microcontrolador para administrar e integrar los módulos GNSS, GSM/GPRS y Tarjeta SD.
- Diseñar una interfaz web Software mediante PHP, Google maps, HTML/JS para la visualización de posición en tiempo real del dispositivo AVL.

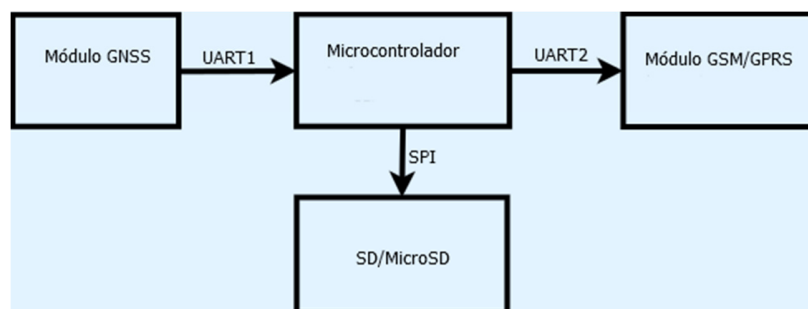
## Capítulo 1 Tecnologías del sistema AVL

### 1.1 Sistema AVL (Rastreo Vehicular Automatizado)

Un sistema AVL (Automatic Vehicle Location) necesita funcionar conjuntamente con otras tecnologías; en este caso los sistemas GNSS (GPS/GLONASS) y una conexión de datos mediante GSM/GPRS con la red celular, además de un servidor web para almacenar los datos y visualizar el tracking en tiempo real con ayuda de mapas (en nuestro caso GOOGLE MAPS).

Un dispositivo de rastreo vehicular (dispositivo transceptor) de manera general está compuesto de un receptor GNSS (GPS/GLONASS) quien obtiene los datos de geolocalización, un transceptor GSM/GPRS para enviar/recibir datos/ordenes al servidor mediante SMS, una SD card para almacenar/respaldar datos localmente, un procesador (Microcontrolador) que se encargue del manejo y control de los módulos mediante la implementación de algoritmos de control que garanticen un correcto funcionamiento del sistema con todos sus elementos [7]. En la figura 1.1 se presenta un diagrama general de un dispositivo AVL.

Comúnmente los módulos GNSS entregan los datos por una interfaz serial (TTL UART) [8] y usan el protocolo NMEA o SIRF binario, algunos dan la opción de poder ser controlados con interfaz I2C/SPI. Con respecto a los módulos GSM/GPRS la mayoría poseen interfaz serial (TTL UART) y comandos AT, en el caso de las SD card se controlan mediante interfaz SPI.

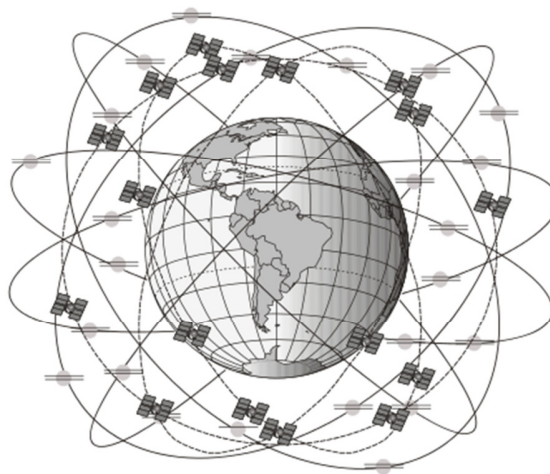


**Figura 1.1** Bloques que integran el dispositivo AVL.

## 1.2 GNSS (Sistema Global De Navegación Por Satélite)

Sistema global de navegación por satélite (GNSS) es un sistema compuesto de satélites que transmite señales utilizadas para el posicionamiento y localización en cualquier parte del globo terrestre, en la figura 1.2 se ilustra la constelación de satélites GPS + GLONASS simultáneamente. Estos permiten determinar las coordenadas geográficas, altitud y otros datos de un punto dado como resultado de la recepción de señales provenientes de constelaciones de satélites artificiales de la Tierra para fines de navegación, transporte, geodésicos, militar, etc. Los más importantes son: GPS (EEUU), GLONASS (RUSIA), GALILEO (EU), BeiDou (CHINA). [9]

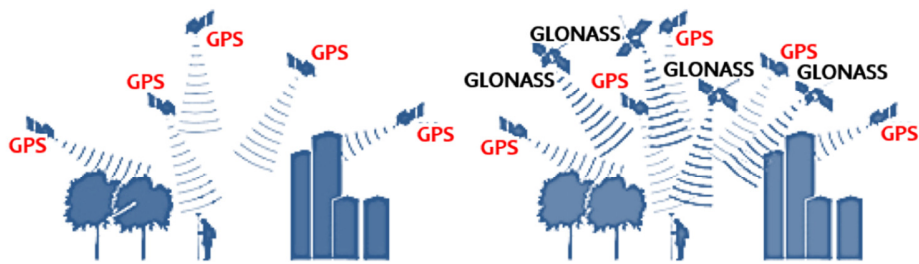
En comparación con usar únicamente GPS, activar múltiples sistemas GNSS generalmente incrementa el número de satélites visibles, reduce el tiempo de la primera posición válida e incrementa precisión, especialmente cuando se mueve sobre ambientes urbanos, en la figura 1.3 se puede apreciar conceptualmente como un dispositivo AVL tiene más satélites en vista usando GPS + GLONASS.



**Figura 1.2** Satélites GPS + GLONASS en órbita.<sup>1</sup>

---

<sup>1</sup> <https://www.researchgate.net>



**Figura 1.3** Vista de satélites GPS y GLONASS.<sup>2</sup>

### 1.3 GPS (Sistema De Posicionamiento Global)

El Sistema de Posicionamiento Global (GPS) es un sistema de navegación basado en satélites que fue desarrollado por el Departamento de Defensa de EE.UU. Es un sistema de doble uso que se puede acceder tanto por militares como civiles. Para civiles la frecuencia de la señal 1575,42 MHz denominada L1, para militares la frecuencia de la señal 1227,60 MHz denominada L2, la cual está cifrada [6]. Se basa en una constelación de veinticuatro satélites que orbitan la tierra a más de veinte mil kilómetros de altura aproximadamente [10].

El Sistema de Posicionamiento Global consiste en tres segmentos principales:

- El segmento espacial.
- El segmento de control.
- El segmento usuario.

**El segmento espacial** está constituido por la constelación NAVSTAR que es un conjunto de satélites que orbitan alrededor de la Tierra como se representa en la figura 1.4. Los satélites GPS se encuentran en seis planos orbitales con tres o cuatro satélites operativos en cada plano. La capacidad operativa de la constelación requiere de 24 satélites pero pueden operar más. [10]. En teoría, se necesitan sólo tres satélites que simultáneamente dan seguimiento a tres distancias de modo que se forman tres esferas. En este caso, el receptor se encuentra en la intersección de las tres esferas, cada distancia es el radio que hay entre el satélite y el receptor, además están centrados en un satélite en particular. Sin embargo, es necesario de un cuarto satélite para tener en cuenta la compensación del reloj en el receptor [10].

<sup>2</sup> <http://inomobile.com.sg>



**Figura 1.4** Constelación de satélites GPS.<sup>3</sup>

Se están desarrollando actualmente varios sistemas de aumentación regional (corrección de error, precisión) como parte de los Sistemas de Satélite de Navegación Global mundial (GNSS por sus siglas en inglés). En los Estados Unidos está el sistema regional basado en GPS llamado Sistema de Aumentación del Ancho de Área (WAAS por sus siglas en inglés) que abarca América del Norte con la posibilidad de extenderse para incluir América del Sur. En Europa se está desarrollando un sistema regional similar llamado Sistema Europeo de Superposición de Navegación Geoestacionario (EGNOS) que cubre Europa y norte de África con la posibilidad de ampliarse e incluir toda África y Oriente Medio [11].

**El segmento usuario** incluye todos los usuarios militares y civiles. Con un receptor GPS conectado a una antena GPS, un usuario puede recibir las señales GPS, que se puede utilizar para determinar su posición en cualquier parte del mundo [11]. Los usuarios pueden adquirir cualquier receptor GPS disponible comercialmente. Este receptor se puede utilizar en vehículos, aviones, equipos telefónicos, barcos, estaciones militares a como se ilustra en la figura 1.5.

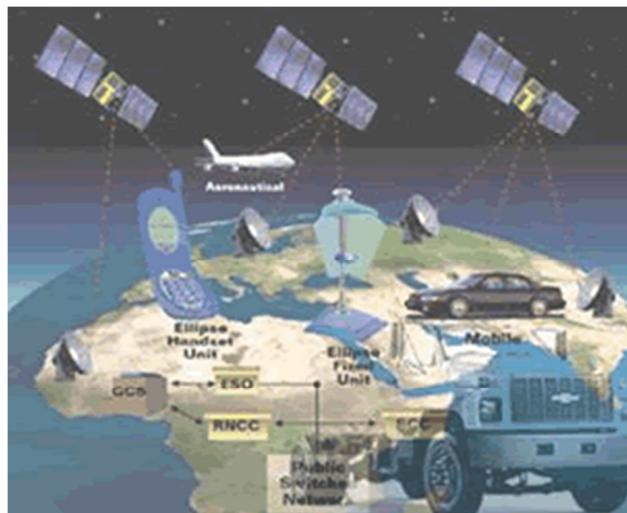
---

<sup>3</sup> <http://rover.info/2060344>

El punto de posicionamiento del GPS, también conocido como posicionamiento independiente o autónomo, implica un solo receptor GPS. Es decir, un receptor GPS rastrea simultáneamente cuatro o más satélites GPS para determinar sus propias coordenadas con respecto al centro de la tierra. Casi todos los receptores GPS disponibles en el mercado actual son capaces de mostrar sus coordenadas para el punto de posicionamiento [11].

**El segmento de control** del sistema GPS se encarga de enviar información a los satélites para controlar las órbitas y realizar el mantenimiento de toda la constelación. El segmento consta de cinco estaciones de control terrestre, incluyendo una estación de control maestro (MCS) ubicada en Colorado Springs, Colorado, E.E.U.U. [11] Las otras cuatro estaciones de monitoreo están situadas en lugares estratégicos en todo el mundo.

Los módulos receptores GPS extraen los datos de posicionamiento, velocidad, tiempo y fecha en mensajes codificados mediante unos estándares. El más conocido se llama NMEA 0183 lo que permite compatibilizar entre otras marcas, pero normalmente también puede ser de tipo binario que depende del fabricante del receptor. [12]



**Figura 1.5** Sistema GNSS con receptores GPS.<sup>4</sup>

---

<sup>4</sup> <http://www.rmasc.com>



## 1.4 NMEA (National Marine Electronic Association).

En los años 1980, NMEA creó el interface único para el intercambio de datos digitales entre los diferentes aparatos utilizados en la marina. El estándar de interface NMEA 0183 es muy aceptado por los fabricantes y agencias marítimas en el mundo. El estándar ha estado adoptado por el comité internacional de electrotécnica de la comunidad Europea como la base para el estándar internacional. [13]

En el estándar NMEA 0183 se utiliza un simple protocolo de comunicaciones serie ASCII, que define cómo los datos se transmiten en una "sentencia" de un "dispositivo transmisor" a varios "dispositivos receptores" al mismo tiempo. A través del uso de multiplexores intermedios, un transmisor puede tener una comunicación unidireccional con un número casi ilimitado de receptores con el uso de multiplexores, múltiples sensores pueden comunicarse a un único puerto de computadora [13]. En un mensaje/trama NMEA todos los campos están separados por comas y tiene la siguiente forma:

**\$ | *Talker ID* | *MSG ID* | *DATA* | \* | *CHECKSUM* | <CR><LF>**

**<\$>** Carácter de inicio. Toda trama inicia con el carácter <\$>.

**Talker ID** identifica el origen de la trama:

**<GP>** para el receptor GPS.

**<GN>** para el receptor GPS + GLONASS o GPS + GALILEO.

**<BD>** para el receptor GPS + BeiDou.

**<P>** mensaje propietario.

**MSG ID** identifica el tipo de trama. (RMC, GGA, GLL, etc.)

**DATA** información entregada por la respectiva trama.

**<\*>** Carácter final campo de datos.

**CHECKSUM** (1 byte, hexadecimal) XOR de todos los caracteres de la trama entre <\$><\*>

**<CR><LF>** todas las tramas finalizan con CR (0x0D), LF (0x0A).

Ejemplo de tramas NMEA:

**\$GPRMC,041139.000,A,3110.2908,N,12123.2348,E,0.44,128.49,300709,,,A\*65**

**\$GPGGA,041140.000,3110.2907,N,12123.2353,E,1,5,1.42,58.7,M,8.0,M,,\*57**

De la primera trama podemos extraer los siguientes datos:

Tiempo UTC: **04:11:39**, Dato Valido: '**A**', Latitud en grados y minutos: **31° 10.2908'**, Longitud en grados y minutos: **121° 23.2348'**, Velocidad sobre tierra: **0.44** knots, Curso sobre tierra: **128.49°**, Fecha: **30-07-09**(09 -> 2009)

En el caso de la latitud y longitud el módulo la envía en grados y minutos.

Para latitud: **ddmm.mmmm** y para longitud: **dddmm.mmmm**. Debemos convertirla únicamente a grados para poder insertarla fácilmente en googlemaps, para la conversión debemos usar la ecuación 6.1 con lo cual el resultado sería; Latitud **31.1715°**, Longitud **121.3872°**.

$$(\text{grados}) + \frac{\text{minutos}}{60} \quad (6.1)$$

Al resultado en grados se le coloca signo negativo (-) si el indicador es **S** (SUR) y **W** (OESTE). En caso positivo **N** (NORTE) y **E** (ESTE) se deja simplemente como está.

## 1.5 GLONASS (ГЛОНАСС, Global Navigation Satellite System)



**Figura 1.6** Constelación de satélites GLONASS.<sup>5</sup>

Es un sistema de navegación satelital que proporciona el servicio de radionavegación satelital, es administrado por la federación Rusa específicamente por ROSCOSMOS y es usado tanto por civiles como por las fuerzas aeroespaciales rusas. Este provee una alternativa al GPS y es el segundo sistema de navegación satelital operando actualmente con cobertura global y precisión comparable al GPS. [14], en la figura 1.6 la constelación correspondiente al sistema GLONASS.

El desarrollo de GLONASS inició en la unión soviética en 1976, la constelación de satélites fue completada en 1995 y desde el 2010 alcanzó 100% cobertura global. El Sistema GLONASS consiste de tres segmentos principales; El segmento espacial, segmento de control, segmento usuario. Los satélites transmiten 2 tipos de señales: *open standard-precision signal* L1OF/L2OF, y *obfuscated high-precision signal* L1SF/L2SF, señales civiles y militares respectivamente.[15]

La banda L1 es de uso civil y son señales FDMA con frecuencia central de **1602.0 Mhz + n × 0.5625 MHz** donde n es el número de canal de frecuencia satelital, la L2 de uso militar también es una señal FDMA con frecuencia central de **1246 Mhz + n×0.4375 MHz**. Actualmente GLONASS está migrando de señales FDMA a CDMA. [15]

---

<sup>5</sup> <http://rover.info/2060344>

## 1.6 GSM/GPRS.

En la década de los noventa, las redes de telefonía móvil se encontraban en su Segunda generación. Esta se caracterizaba por usar sistemas digitales, frente a los analógicos que se venían usando hasta ese momento. Se conseguía de esta forma mejorar la calidad de la señal de voz y dar la posibilidad de transmitir paquetes de datos. El estándar que se impuso en Europa, y sigue usándose en la actualidad, es GSM. [16], en la figura 1.7 se puede apreciar un diagrama general de la red GSM, el dispositivo GSM se comunica directamente con la estación base, desde donde se obtiene acceso a otras redes mediante el sistema de administración de red y los subsistemas de red.

La red GSM preveía unos servicios de transmisión de datos desde su fase inicial. Sin embargo, se trataba de servicios con modalidad de transferencia por conmutación de circuitos. Esto quiere decir que la red, una vez establecida la conexión física entre dos usuarios, dedicaba los recursos propios hasta que no fuera solicitada expresamente la liberación de la conexión, independientemente del hecho de que los dos usuarios se intercambiarían datos o no durante todo el proceso de conexión. [16]

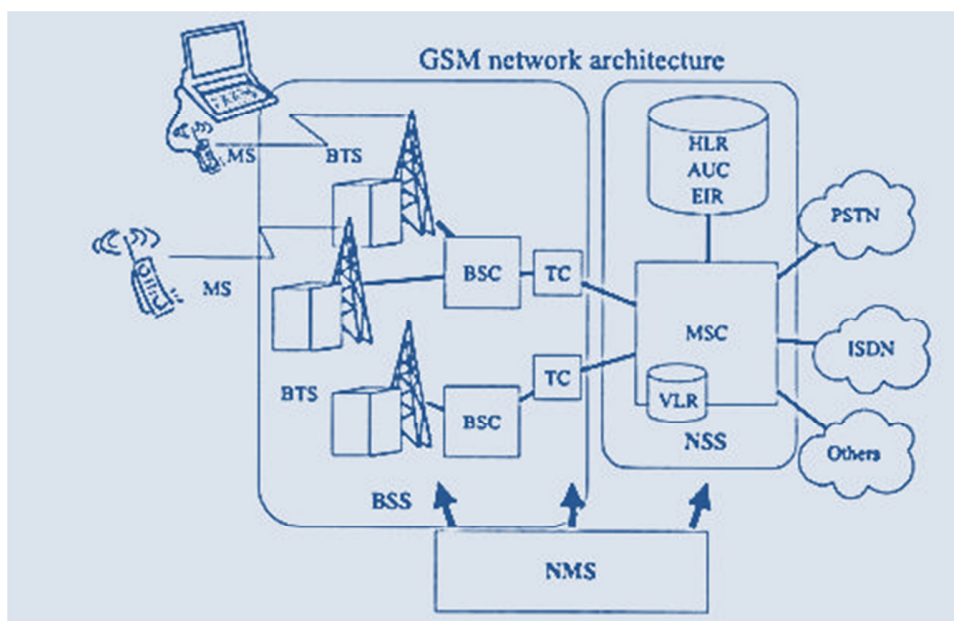


Figura 1.7 Red GSM [13]

Los objetivos del Sistema General de Paquetería por Radio (GPRS) es el de permitir la transmisión de paquetes de datos (tener la comunicación siempre activa, y el uso del canal de radio solo cuando lo necesitamos), controlar la tasa de datos en la interfaz de radio. El GPRS es definido como una parte del sistema GSM y está adaptado a él. [17]

El servicio GPRS requiere algunos cambios en los elementos de la red de radio como en la estación base y en los controladores de la estación base. Estos cambios son principalmente software para la estación base y software/hardware para los controladores de la estación base y sus funcionalidades/limitaciones son consideradas cuando se dimensiona la red de radio. [17]

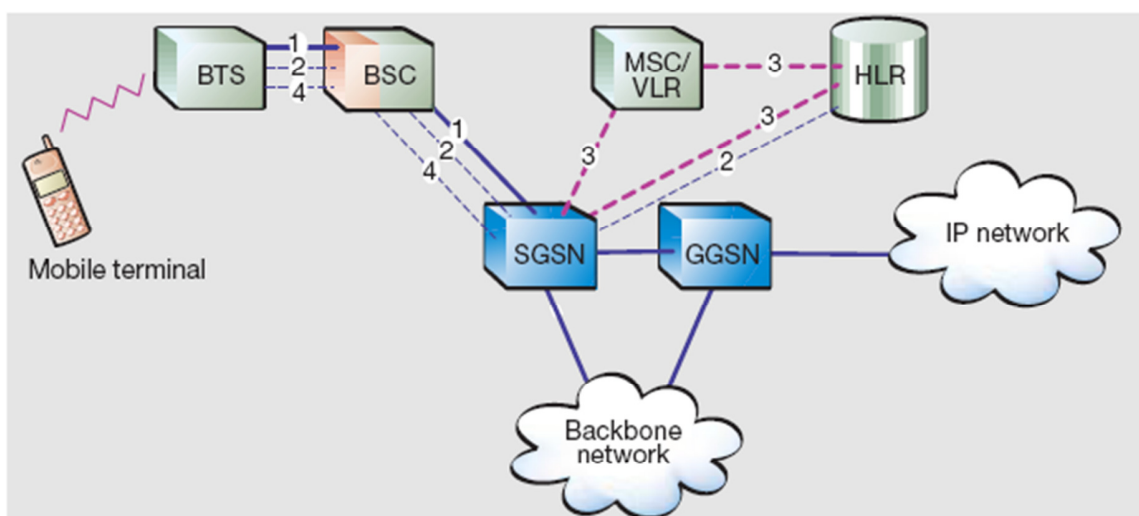
La BSS relacionada a la implementación de GPRS implica cambios en la red de radio y la instalación de nuevos elementos llamados SGSN (Serving GPRS SupportNode), GGSN (Gateway GPRS SupportNode), etc.

El **SGSN** (Serving GPRS SupportNode) su función principal es autenticar un terminal al momento de realizar una conexión de datos. Se encarga de la parte movilidad del celular además de dar acceso a estos a la red de datos móviles, es uno de los componentes principales de la red GPRS que maneja el enrutamiento de paquetes de datos dentro de la red celular (Administración móvil, Autenticación de usuarios y asignar la calidad del servicio a utilizar por cada terminal). [17]

El **GGSN** (Gateway GPRS SupportNode) es uno de los componentes principales de la red GPRS. El GGSN es responsable por la interconexión entre la red GPRS y redes externas conmutadas por paquetes e Internet. Se encarga de proveer la salida a la PDN (Packet Data Network) dependiendo del APN, así como de la parte de facturación y aplicación de políticas y reglas de navegación. [18]

Desde el punto de vista de redes externas, el GGSN es un router a una sub-red, porque el GGSN 'esconde' la infraestructura GPRS de la red externa. Cuando el GGSN recibe dato direccionado a un usuario específico, este revisa si el usuario está activo. Si está activo el GGSN envía el dato al usuario móvil, pero si el usuario móvil está inactivo, el dato es descartado. Al otro lado los paquetes originados por el usuario (dispositivo móvil) son direccionados a la red correcta por el GGSN. Al hacer todo esto el GGSN mantiene un record de los usuarios móvil activos, permite direccionamiento IP y también es responsable de la facturación. [18]

En la figura 1.8 podemos ver los subsistemas que facilitan la conexión GPRS, el dispositivo se conecta directamente a la estación base, luego el servidor SGSN válida la conexión con ayuda de registros para posteriormente enlazar con el GGSN el cual es el Gateway entre la red móvil y la red IP.



**Figura 1.8** Conexión GPRS [19]

## 1.7 Comandos AT

Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre usuario y un terminal. El juego de comandos AT fue inicialmente desarrollado por Dennis Hayes como una interfaz de comunicación para configurar módems y proporcionar instrucciones, actualmente la telefonía móvil GSM lo ha adoptado como lenguaje estándar para poder comunicarse con sus terminales. [20]

Todos los teléfonos móviles GSM poseen un juego de comandos AT específico que sirve de interfaz para configurar y proporcionar instrucciones a los terminales. Este juego de instrucciones puede encontrarse en la documentación técnica de los terminales GSM y permite acciones tales como realizar llamadas de datos o de voz, leer y escribir en la agenda de contactos y enviar mensajes SMS, además de muchas otras opciones de configuración del terminal.[21]

Mediante los comandos AT controlamos el módulo, nos comunicamos con él y le damos instrucciones. Los comandos AT permiten realizar llamadas, enviar SMS, consultar los parámetros del módulo, medir la señal celular y realizar todas las configuraciones del SIM900. [22] La Comunicación es de tipo conversación. Siempre se debe esperar la respuesta antes de enviar el próximo comando. Al emitir un comando AT este debe finalizar con CR (0x0D), LF (0x0A). [23]

## 1.8 Microcontrolador

Un microcontrolador es un circuito integrado digital que puede ser usado para diversos propósitos debido a que es *programable*. En su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada/salida y periféricos. Estas partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se le conoce como microcomputadora. Un microcontrolador tiene los mismos bloques de funcionamiento básicos de una computadora lo que nos permite tratarlo como un pequeño dispositivo de cómputo. [24]

## **Universal Asynchronous Receiver Transmitter (UART)**

Es uno de los periféricos disponibles en los microcontroladores. El UART es un transceptor serial asincrónico que comúnmente usa el protocolo RS232 (TTL) para enviar y recibir datos. Los puertos seriales son muy comunes y populares, además la manera más sencilla de transmitir información en modo serial. Para las familias de microcontroladores que hemos seleccionado las tasas de transferencia van de los 110 BPS hasta los 500K BPS. El UART solo usa 2 pines para transmitir/recibir información Tx/Rx respectivamente aunque también puede ser usado de manera sincrónica. [24]

## **The Serial Peripheral Interface (SPI)**

Es una interfaz serial síncrona útil para comunicar con otros periféricos o microcontroladores a velocidades bastante altas. El SPI usa 4 pines: SDO (salida digital serial), SDI (entrada digital serial), SCK (señal de reloj), SS (selección de dispositivo en modo esclavo). [24] En nuestro caso lo utilizamos para controlar la SDCard.

## **Pines mapeables**

Esta es una característica de las familias de microcontroladores que seleccionamos la cual consiste en la libertad de elegir en que pines colocamos por ejemplo los periféricos SPI, UART, PWM, etc. Los pines mapeables son 16 y proporcionan una excelente flexibilidad para diseños de propósito general, lo cual ha permitido en gran parte que el circuito diseñado funcione con múltiples integrados de MICROCHIP con similares características.

## **Herramientas software/hardware.**

### **MPLAB X IDE**

Es un programa/software interfaz el cual es usado para desarrollar (código) aplicaciones con los microcontroladores y controladores digitales de señal (DSC) de MICROCHIP.

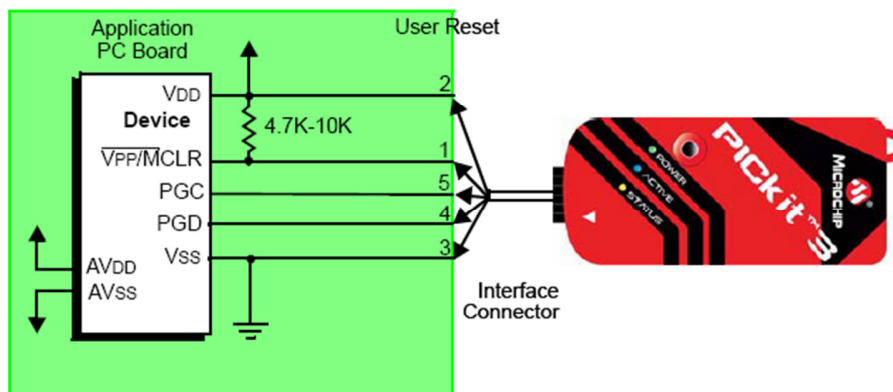


## Compilador MPLAB XC16

Es un compilador de lenguaje C multiplataforma (Windows, Linux) para todos los dispositivos MICROCHIP de 16 Bits (PIC24/dsPIC). [25]

## PICKit 3 (Programador/Depurador)

Es un programador/depurador en circuito esto significa que no es necesario extraer el MCU de la tarjeta de aplicación para programar y depurar. Este aparato de bajo costo sirve para grabar (no producción) la mayoría de dispositivos que estén basados en ICSP (Programación Serial en Circuito) de las familias PIC12, PIC16, PIC18, PIC24F/H, dsPIC30/33, etc. Por este método solo se requiere conexión con 3 pines del dispositivo (PGD, PGC, MCLR) aparte de VCC y GND [26]. En la figura 1.9 se observa la conexión ICSP del dispositivo grabador con el microcontrolador seleccionado.



**Figura 1.9** Conexión ICSP – PICKIT3. [46]

## 1.9 Secure Digital SDC

Secure Digital (SD) es un estándar de tarjeta de memoria para dispositivos portátiles siendo actualmente uno de los formatos comúnmente utilizados. El estándar SD fue desarrollado por empresas privadas e introducido como una mejora evolutiva de las tarjetas MMC. El estándar es mantenido por la Asociación de Tarjetas SD en la que participan varios fabricantes. [27]

El estándar es propietario y requiere licencia, pero hay un sub-conjunto de especificaciones SD que han sido liberadas al público por la SDCA (SDC Association) en la forma de “Especificación física simplificada”. [27], en la figura 1.10 se observa la numeración y nombramiento de pines para las tarjetas SD y microSD.

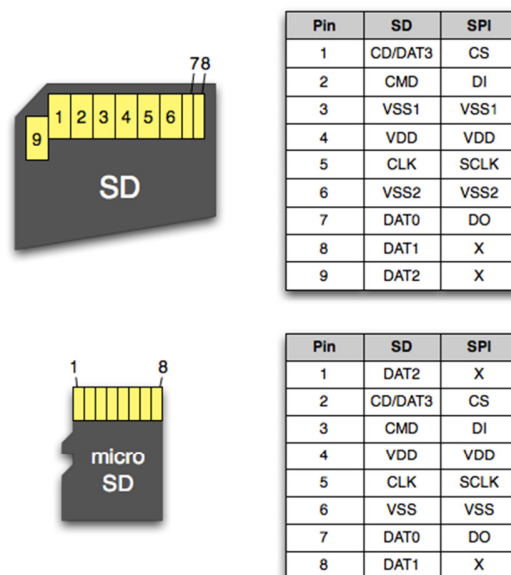


Figura 1.10 SD/ MicroSD pines.<sup>6</sup>

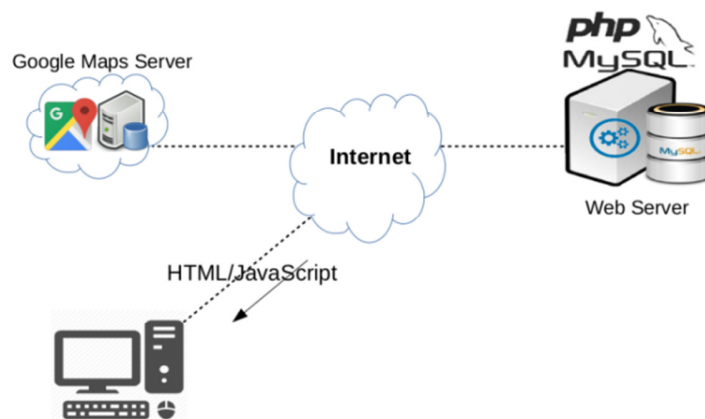
<sup>6</sup> <http://www.elasticsheep.com>

### 1.10 Interfaz de usuario

La interfaz de usuario es el medio con que el usuario puede comunicarse/interactuar con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo. Normalmente suelen ser fáciles de entender y fáciles de accionar.

Las interfaces básicas de usuario son aquellas que incluyen elementos como menús, ventanas, teclado, ratón y algunos otros sonidos que la computadora hace, y en general, todos aquellos canales por los cuales se permite la comunicación entre el ser humano y la computadora.

Nuestra interfaz es una página web la cual brinda el servicio de tracking. El usuario con ayuda de Google maps embebido en la web, puede interactuar con un mapa y ver la trayectoria del dispositivo AVL, en la figura 1.11 se aprecia un diagrama de la conexión entre usuario-servidor.



**Figura 1.11** Conexión al Sitio WEB.

## **1.11 Tecnologías Sitio Web.**

### **1.11.1 Cookies**

Para el desarrollo de aplicaciones web se necesita frecuentemente mantener estados. Para esto se usan las cookies, que es información que un servidor puede almacenar en el sistema cliente. Esto le permite a las aplicaciones web instituir la noción de "sesión", y también permite rastrear usuarios ya que las cookies pueden guardarse en el cliente por tiempo indeterminado [28]. Por ejemplo una cookie de google la cual puede funcionar para múltiples propósitos; seguridad, procesos, preferencias, publicidad, inicio de sesión.

### **1.11.2 HTML / CSS (HyperText Markup Language)**

HTML (HyperTextMarkupLanguage) es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web.[29]

CSS (Cascading Style Sheets) es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML, La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. La información de estilo puede ser definida en un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo «style». [19]

### **1.11.3 PHP (Hypertext Pre-processor)**

PHP es un lenguaje de programación de uso general, el código PHP se ejecuta en un servidor web con un motor de procesamiento PHP que genera la página Web resultante. Este fue originalmente diseñado para el desarrollo web de contenido dinámico. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo. [30]

El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable. Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos. [30]

#### **1.11.4      JavaScript / AJAX**

JavaScript es un lenguaje de programación interpretado, se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.[31]

El uso más común de JavaScript es escribir funciones embebidas o incluidas en páginas HTML y que interactúan con el DocumentObjectModel (DOM) de la página.

AJAX (Asynchronous JavaScript And XML) es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones. Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML. Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y DocumentObjectModel (DOM). [31]

### **1.11.5 Google Maps API**

Google Maps es un servicio de mapas web o herramienta de vista geoespacial. La API consiste de librerías de clases y métodos con los cuales se puede desarrollar una aplicación interactiva.[32] Una API define una manera estándar para un programa de cómo llamar funcionalidades/código que está dentro de otra aplicación o librería. La API Google define un conjunto de objetos y métodos JavaScript/AJAX que el usuario puede usar para colocar mapas en sus propios sitios web.[33]

La API es soportada por los más importantes navegadores WEB y nos brinda una serie de funcionalidades; Eficiente renderizado de mapas, geocoding, routing, objetos configurables, interacción con mouse y touch(móvil), controles UI pre-construidos, etc. [34]

### **1.11.6 Base de Datos**

Una base de datos (BD) es un conjunto integrado de datos no repetidos que modelizan un universo dado, almacenados en un soporte informático, organizados de forma independiente de su utilización y accesibles simultáneamente por distintos usuarios y aplicaciones. [35]

También se puede decir que una base de datos es un contenedor que almacena tablas y otras estructuras SQL relacionadas a esas tablas. [36]

Es decir, la diferencia de una BD respecto a otro sistema de almacenamiento de datos es que éstos se almacenan de forma que cumplan tres requisitos básicos:

- ❖ No redundancia: Los datos se almacenan una sola vez.
- ❖ Independencia: Los datos se almacenan teniendo en cuenta la estructura inherente a los propios datos y no la de la aplicación que los crea. Se puede hablar de dos tipos de independencia: independencia física, de tal manera que la estructura física de la BD puede ser modificada de forma transparente para los programas que la utilizan, e independencia lógica, es decir el programador usa la Base de Datos, pero desconoce su estructura interna
- ❖ Concurrencia: Varios usuarios, ejecutando la misma o diferente aplicación, podrán acceder simultáneamente a los datos.

SQL (Structured Query Language) es un lenguaje estándar ANSI/ISO de definición, manipulación y control de bases de datos relacionales. Declarativo (sólo hay que indicar qué se quiere hacer) de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ellas. El SQL es un lenguaje con el que se puede acceder a todos los sistemas relacionales comerciales. [37]

MySQL es un sistema (software libre.) de administración/gestión de bases de datos relacional, multi-hilo. Este almacena la información del usuario y se puede interactuar con él utilizando SQL. [38]

### **1.11.7 HTTP (HyperText Transfer Protocol)**

Es un protocolo de la capa de aplicación para sistemas de información hipermedia, distribuido y colaborativo. Este es un protocolo genérico, sin estado (es decir, no guarda ninguna información sobre conexiones anteriores) que puede ser usado para muchas tareas más allá de su uso para hipertexto. HTTP ha estado en uso por la WWW (World Wide Web) desde 1990. [39]. Algunas de sus versiones son HTTP/1.0, HTTP/1.1.

Una transacción HTTP está formada por un encabezado seguido, opcionalmente, por una línea en blanco y algún dato. El encabezado especificará cosas como la acción requerida del servidor, el tipo de dato de retorno, o el código de estado.

El uso de campos de encabezados enviados en las transacciones HTTP le da gran flexibilidad al protocolo. Estos campos permiten que se envíe información descriptiva en la transacción, permitiendo así la autenticación, cifrado e identificación de usuario.

HTTP define algunos métodos que indican la acción que desea que se efectúe sobre el recurso identificado. Los métodos que nos interesan para este proyecto son GET/POST.

**GET:** Pide una representación del recurso especificado.

**POST:** Envía los datos para que sean procesados por el recurso identificado. Los datos se incluirán en el cuerpo de la petición.

Al enviar una petición al servidor éste envía un código de respuesta, algunos de los más comunes se aprecian en la tabla 1.1. Para la lista completa revisar HTTP1.1 (RFC2616).

**Tabla 1.1** Código respuesta HTTP

Número de Código	Descripción.
200	Operación exitosa.
400	Solicitud incorrecta.
401	No autorizado.
404	No encontrado.
601	Error de RED.



## Capítulo 2 Diseño del Sistema

### 2.1 Subsistema GNSS

Este subsistema se encarga de Recibir y descifrar las señales provenientes de los satélites que brindan el servicio de geo-localización, este subsistema está compuesto por una antena activa y un módulo receptor.

Al seleccionar el módulo GNSS teníamos en cuenta que debía ser de fácil implementación. Las opciones más accesibles con costos similares son presentadas en la tabla 2.1.

**Tabla 2.1** Lista de módulos evaluados.

	<b>PMB-688/L10</b>	<b>L76-M33</b>	<b>L16/L26</b>	<b>NEO-7M/N</b>
<b>GNSS</b>	GPS	GPS/GLONASS + GALILEO	GPS/GLONASS	GPS/GLONASS + GALILEO

Todos estos módulos GNSS son fácilmente integrables con cualquier proyecto, entre estos módulos podríamos decir que la mejor opción por sus características son el L76 y el L16/L26 (QUECTEL) estos poseen una excelente inmunidad al ruido y funcionan simultáneamente con GPS (CDMA) y GLONASS (FDMA).

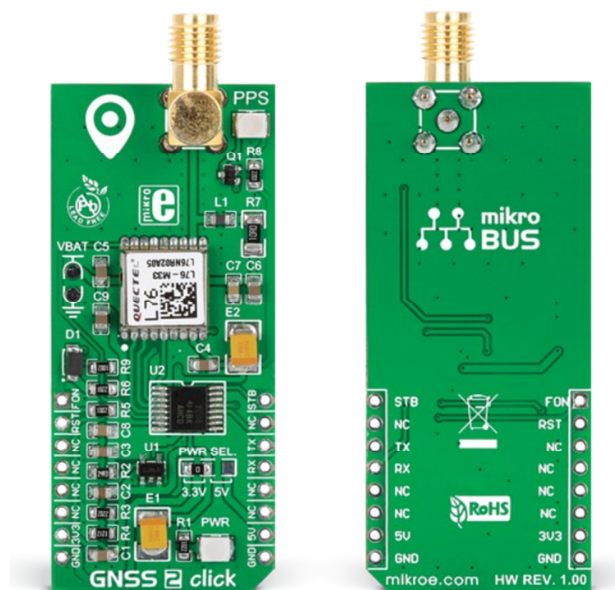
El L76 es un módulo GNSS que posee excelente protección contra EMI y otras fuentes de ondas electromagnéticas como Wi-Fi, GSM, Bluetooth, etc. Actualmente hay 2 versiones del dispositivo; L76 (GPS/GLONASS) y L76B (GPS/BEIDOU), próximamente al estar operativo el sistema GALILEO saldrá el L76G (GPS/GLONASS/GALILEO), [40] es decir que nuestro sistema usando L76 está en capacidad de ser actualizado para soportar un sistema GNSS mas, el cual sería GALILEO.

La Segunda opción es el NEO-7M/N, éste módulo usa GPS/GLONASS y hasta GALILEO, pero lamentablemente solo un sistema puede ser usado a la vez lo que le quita el atractivo.

Las otras opciones son el L10 y el PMB-688. Entre estos dos el L10 es más resistente a interferencias que el PMB-688. Nosotros usamos el L16 y L76 pero el código driver que desarrollamos funciona también para el L10 y L26.

El L16 fue pensado para poder ser usado con otras tecnologías siendo así resistente a la interferencia de señales WiFi, WiMax, CDMA, GSM. El L16 usa dos protocolos: el NMEA 3.0 para entregar información y el PMTK (protocolo Propietario) para notificar y recibir instrucciones. [41]

EL módulo L16 de PANDATRON viene en un pequeño PCB únicamente con un conector SMA (Male) para la antena, la alimentación se hace por los pines GND y VCC (3V3), el envío y recepción de tramas se hace mediante los pines UART TTL (TX, RX).



**Figura 2.1** Módulo MIKROE-1887 con L76-M33 Quectel.<sup>7</sup>

<sup>7</sup> <http://www.mikroe.com/click/gnss2/>

El módulo GNSS L76 se alimenta con 3.3V y basta con alimentarlo para que empiece a enviar tramas NMEA pero deben pasar un par de segundos antes que el L76 envíe tramas validas con los datos reales en donde se encuentra el aparato. Para el L76 son: COLD START <35 seg, HOT START < 1 seg [42]. Después de esto puede enviar datos al microcontrolador con frecuencia de hasta 10 Hz (cada 100ms) aunque en nuestro caso es 1 seg, en la figura 2.1 se aprecia un módulo MIKROE-1887 el cual usa L76-M33.

Para configurar el aparato se le envía ordenes mediante tramas MTK protocolo privado; PMTK con las cuales se puede establecer el BaudRate de la comunicación, desactivar tramas innecesarias establecer la frecuencia de algunos tipos de datos por ejemplo RMC (cada segundo), GGA (cada 2 segundos). El módulo MIKROE-1887 cuyo circuito aparece en la figura 2.2 recibe y envía datos en RS232 (TTL 5v/3.3v) esto con ayuda de un circuito aislador, en la figura 2.3 se aprecia la conexión entre el L76 y el microcontrolador para permitir la comunicación entre ambos.

Las tramas NMEA que posee el L76 Quectel son: RMC, VTG, GGA, GSA, GSV, GLL, ZDA, TXT, etc. Para nuestra aplicación solo necesitamos RMC y GGA, las otras nos son redundantes e innecesarias. De la trama RMC extraemos; tiempo UTC, fecha, latitud, longitud, velocidad sobre tierra, curso sobre tierra, etc. GGA de aquí extraemos altitud sobre el nivel del mar, número de satélites. [43]

Las principales tramas PMTK que usamos son:

PMTK251 – Establece el baudrate para la comunicación.

PMTK314/514 – Desactivar tramas innecesarias, establecer frecuencia de tramas.

Ejemplo trama PMTK:

**\$PMTK251,38400\*27<CR><LF>**

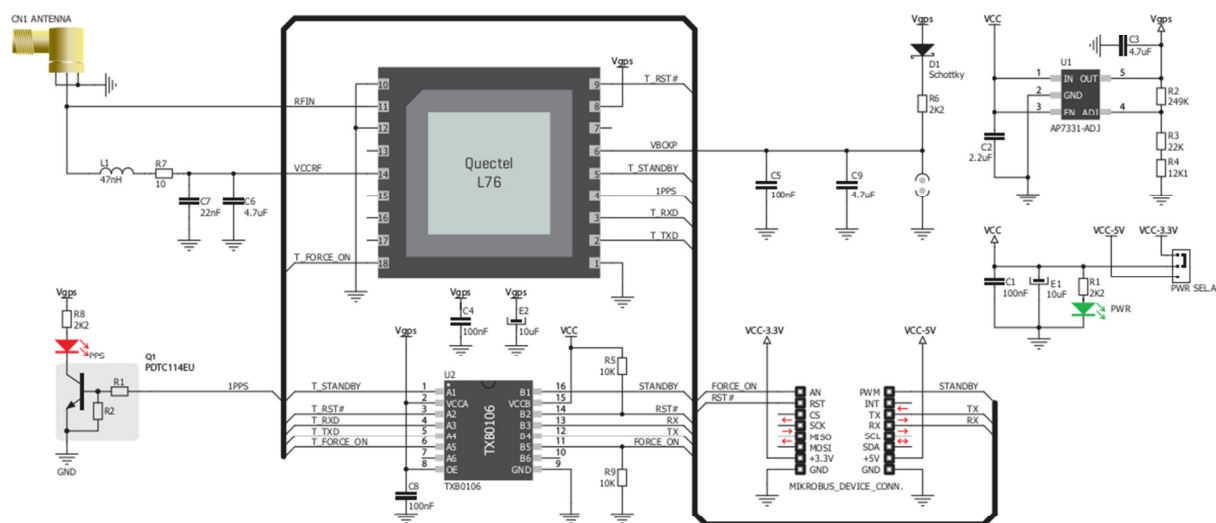


Figura 2.2 Circuito MIKROE-1887 con L76-M33 Quectel.<sup>8</sup>

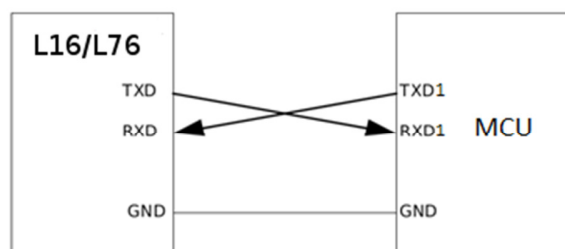


Figura 2.3 Conexión de la interfaz serial L16/L76.

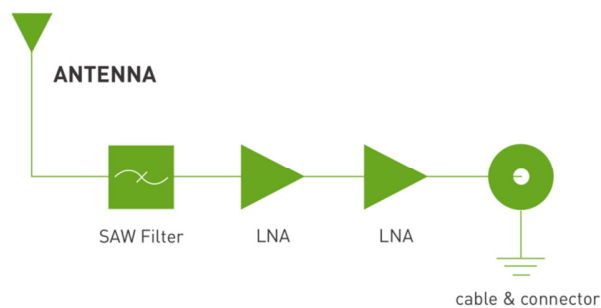
## ANTENA GNSS

Los requerimientos de la antena para este proyecto consisten en que debía ser cableada con conector SMA además de soportar GPS/GLONASS. Para este proyecto usamos la antena activa de montaje magnético GPS/GLONASS **AA.161.301111** la cual se puede apreciar en la figura 2.4 junto a su diagrama de bloques en la figura 2.5 Esta antena viene con un LNA (Amplificador de Bajo Ruido) lo que proporciona una ganancia entre 22 dB – 31 dB. [44]

<sup>8</sup> <http://www.mikroe.com/click/gnss2/>



**Figura 2.4** Antena activa **AA161.301111**. [44]



**Figura 2.5** Diagrama interno, antena activa **AA161.301111**. [44]

Características:

- Frecuencia: 1575 Mhz – 1610 Mhz. (GPS/GLONASS)
- Cable RG174 de 3 Mts.
- Conector: SMA (Male).
- Impedancia: 50 ohm.

## 2.2 Subsistema GSM/GPRS

Existen varias tecnologías (GPRS, EDGE, UMTS, HSPA, etc.) que pueden ser usados para enviar datos al servidor. Para seleccionar la adecuada debemos tomar en cuenta algunos aspectos claves; volumen de datos, costos relacionados/disponibilidad, velocidad de transferencia, etc.

Los módulos que usan UMTS, HSPA, EDGE son considerablemente más costosos que un módulo únicamente GSM/GPRS. En aplicaciones como rastreo vehicular el volumen de datos enviado es bajo en comparación con otros servicios, comúnmente esta demanda de recurso se satisface con GPRS. Por cuestiones de costo, disponibilidad y requerimientos técnicos (volumen de datos) seleccionamos la tecnología GSM/GPRS.

Este subsistema se encarga de conectarse a la red celular mediante GSM/GPRS y de enviar los datos al servidor web mediante TCP/IP, este subsistema está compuesto por una antena y un módulo receptor.

Para nuestro diseño necesitamos un dispositivo GSM/GPRS que sea quad-band y que ya tenga integrado los protocolos TCP/IP, en el mercado hay muchos vendedores que ofertan los módulos ya integrados con ranura SIM card, conector SMA, adaptación de niveles y regulador de voltaje. Los más económicos y fáciles de usar son; *GPRS MODULE v1.0* de ITEAD STUDIO el cual usa **SIM900**, *GSM CLICK* el cual usa un **GL865-QUAD** y el *GSM2 CLICK* el cual usa un **M95** (Quectel) ambos de Mikroelektronika. Entre estos tres aparatos los cuales poseen características similares seleccionamos el SIM900 debido a la extensa documentación y por cuestiones de disponibilidad, sin embargo para diseños posteriores se podría usar el M95 (quectel) ya que tiene mejoras considerables referentes al uso eficiente de energía, protocolos y capacidades en general. Por ejemplo; el M95 soporta SSL para conexión encriptada. El SIM900 necesita una actualización de software para soportarlos. En la tabla 2.2 se aprecia los módulos evaluados.

**Tabla 2.2** Lista de módulos GSM/GPRS evaluados.

	<b>SIM900</b>	<b>GL865-QUAD</b>	<b>M95</b>
<b>GSM/GPRS</b>	Quad-Band	Quad-Band	Quad-Band
<b>SSL</b>	Software Update	-	SSL

## Descripción módulo SIM900.



**Figura 2.6** Módulo SIM900 Quad-Band.<sup>9</sup>

El SIM900 es un módulo Quad-Band GSM/GPRS el cual es controlado mediante comandos AT. Nosotros usamos el **GPRS MODULE v1.0** de ITEAD STUDIO que ya viene con un conector SMA para la antena y ranura para la tarjeta SIM, lo podemos apreciar en la figura 2.6, este se alimenta con 5V y se controla mediante los pines UART TTL (RX, TX) además de los pines RST (para enviar señal de RESET) y PWKY (para Apagar/Encender el módulo), para la comunicación con microcontrolador se debe conectar Rx del módulo con Tx del MCU y Rx del MCU con Tx del módulo, como se observa en la figura 2.7.

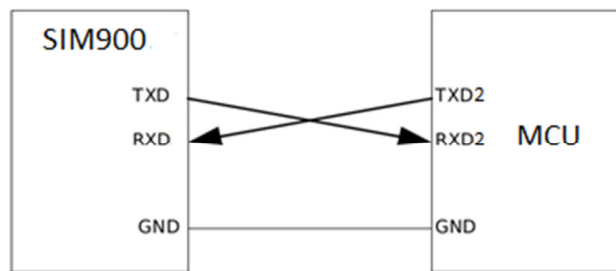
### Características:

- Quad-Band 850/ 900/ 1800/ 1900 MHz.
- GPRS multi-slot clase 10GPRS Móvil estación clase B.
- Control vía AT comandos (GSM 07.07 ,07.05 y SIMCOM AT Comandos).
- TCP/IP embebido, UDP, PPP, FTP, HTTP, etc.
- Bajo consumo: 1.5mA (modo sleep).
- Temperatura de operación: -40 °C a +85 °C.

Al alimentar el SIM900 este necesita recibir un pulso por el pin PWKY para encender, una vez encendido está listo para recibir comandos AT por su interfaz serial. El SIM900 solo procesa un comando AT a la vez, así que si está ocupado procesando algún comando dictado anteriormente y no ha emitido respuesta, el módulo no puede recibir más comandos. En determinadas ocasiones es necesario aplicar RESET por hardware al aparato y esto se logra enviando un pulso en el pin RST (RESET). [45]

---

<sup>9</sup> <https://www.itead.cc/>



**Figura 2.7** Conexión de la interfaz serial SIM900.

### Antena Quad-Band GSM



**Figura 2.8** Antena GSM QuadBand.<sup>10</sup>

Para este proyecto se ha seleccionado una antena cableada (cable de 3 mts de longitud) con conector tipo SMA la cual corresponde a la figura 2.8.

Especificaciones:

- Quad-band: 850/900/1800/1900MHz y 2.4GHz.
- Para GSM/GPRS, CDMA/1X, TDMA/AMPS, CDPD.
- Cable RG174A/U – 3m longitud.
- Conector: SMA (Male).
- Frecuencia: 824 ~ 960 MHz, 1710 ~ 1990 MHz.
- Impedancia: 50 Ohm nominal, VSWR: <2.0: 1.
- Ganancia: 3.5dBi, Radiación: Omnidireccional.
- Onda: Dipolo ½ onda, Polarización: Vertical.
- Dimensiones: 115x22x4mm.

<sup>10</sup> <https://www.sparkfun.com/>



Secuencia de comandos, al encender el módulo se deben enviar los comandos listados en la tabla 2.3.

**Tabla 2.3** Comandos AT de inicio.

Comandos	Respuesta	Descripción
<b>A... AT</b>	OK	Sincronizar y cerciorarse que responde.
<b>AT+CFUN=1</b>	OK	Funcionalidad completa.
<b>AT+CPIN?</b>	+CPIN: READY	PIN ok?
<b>AT+CSQ</b>	+CSQ: rssi, ber	Nivel de la Señal.
<b>AT+CMGF=1</b>	OK	SMS Modo Texto

Para iniciar una conexión GPRS es necesario emitir la siguiente secuencia de comandos presentados en la tabla 2.4.

**Tabla 2.4** Comandos AT inicio GPRS.

Comandos	Respuesta	Descripción
<b>AT+CREG?</b>	+CREG: 0,1	Registrar en la red.
<b>AT+CGATT?</b>	+CGATT: 1	Conectado a servicio GPRS?
<b>AT+CGATT=1</b>	OK	En caso de no estar conectado a servicio GPRS (+CGATT: 0). Conectar.

Una vez recibida respuestas positivas de los comandos anteriores procedemos con los comandos expuestos en la tabla 2.5.

**Tabla 2.5** Comandos AT de conexión GPRS.

Comandos	Descripción
<b>AT+CSTT</b>	Inicia tarea GPRS y define APN, CONTRASEÑA, USUARIO.  En nuestro caso emitimos el comando sin definir nada, es decir así como está. La respuesta esperada es 'OK' y comúnmente es enviada en un par de segundos.
<b>AT+CIICR</b>	Establece/Levanta la conexión inalámbrica GPRS. La respuesta esperada es 'OK' y según las condiciones de la señal/conexión puede tardar bastante tiempo en ser emitida (hasta 60 Seg).
<b>AT+CIFSR</b>	Consigue la dirección IP (dinámica) asignada temporalmente al dispositivo. Este comando debe ser obligatoriamente emitido. La respuesta de este comando es simplemente nuestra dirección IP y es emitida al instante.
<b>AT+CIPSTATUS</b>	Pregunta el estado actual de la conexión. La respuesta de este comando es 'STATE: <estado>'

En la tabla 2.6 se describe en detalle la función del comando CIPSTART la cual es establecer una conexión TCP/UDP con el servidor.

**Tabla 2.6** Comandos AT conexión sitio web.

Comandos	Descripción
<b>AT+CIPSTART="TCP","185.28.21.212",80</b> <b>AT+CIPSTART="TCP","openavl.esy.es",80</b>	El comando CIPSTART inicia conexión UDP/TCP necesita parámetros como número de puerto, dirección IP o dominio al cual conectarse. La respuesta es 'OK' 'CONNECT OK' 'ALREADY CONNECT' 'CONNECT FAIL', etc.  En nuestro caso es una conexión TCP usando el puerto 80 (usado para protocolo HTTP) con la IP estática de nuestro servidor o nuestro dominio.

Los comandos referentes al envío de datos, cierre y desconexión, son descritos en detalles en la tabla 2.7.

**Tabla 2.7** Comandos AT Envío/Cierre/Desconexión.

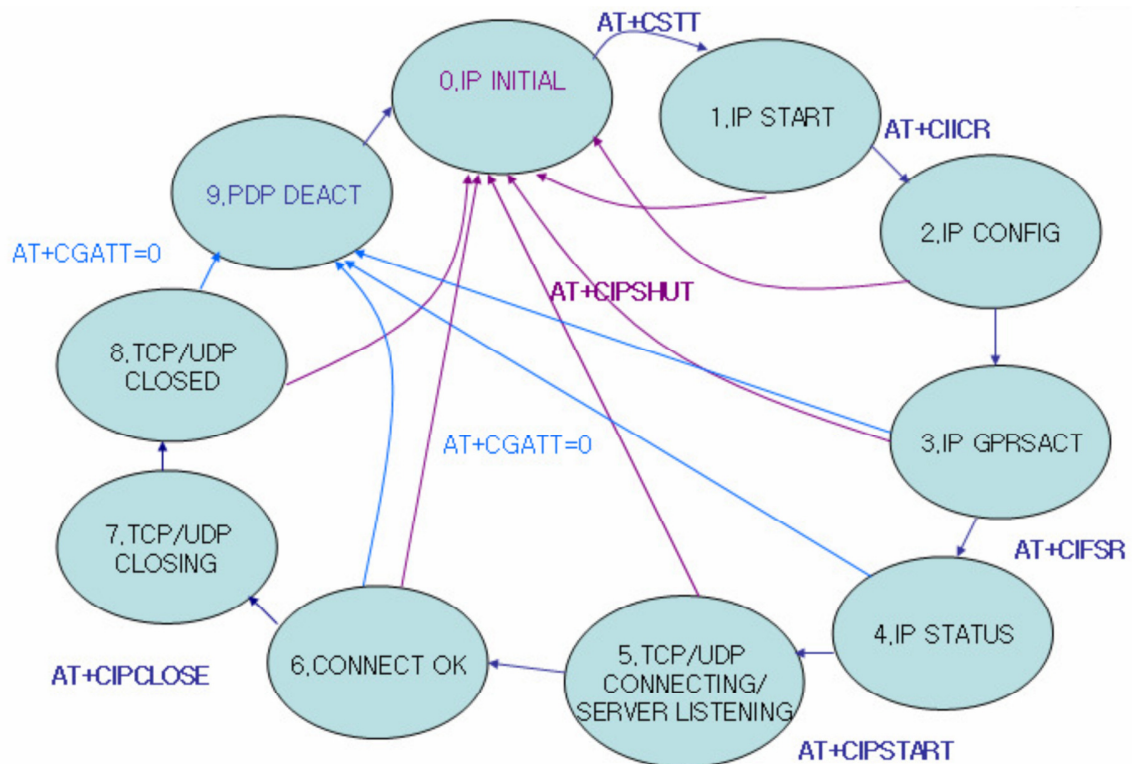
<b>Comandos</b>	<b>Descripción</b>
<b>AT+CIPSEND</b>	Envía dato a través de la conexión TCP/UDP, la correcta ejecución de este comando se hace en 2 partes. Primero se emite el comando así como está y se espera como respuesta el carácter '>', una vez recibido este carácter enviamos el dato que queremos transmitir finalizando con el carácter SUB (0x1A) <ctrl+z>. La respuesta esperada de este comando es un 'SEND OK' además de datos en caso de que el servidor apuntado responda.
<b>AT+CIPCLOSE</b>	Cierra la conexión TCP o UDP. La respuesta esperada a este comando es un 'CLOSE OK'
<b>AT+CIPSHUT</b>	Desactiva conexión actual GPRS. La respuesta esperada a este comando es 'SHUT OK'.

Otros comandos de gran utilidad son descritos en la tabla 2.8.

**Tabla 2.8** Otros comandos AT

<b>Comandos</b>	<b>Descripción</b>
<b>AT+CPOWD=1</b>	Apaga el módulo de manera urgente pero si el comando es 'AT+CPOWD=0' lo apaga de manera normal.
<b>AT+CDNSGIP="www.yandex.com"</b>	Pregunta por la respectiva dirección IP de determinado dominio.
<b>AT+CMGS="+505#####"</b>	Envía mensaje de texto a determinado número. La ejecución de este comando se realiza en dos partes; se emite este comando y se espera por el carácter '>' y una vez recibido se escribe el dato a ser enviado finalizando con el carácter SUB (0x1A) <ctrl+z>.
<b>AT+CMGL=&lt;stat&gt;</b>	Lista los mensajes SMS preferidos. <stat> = 'REC READ', 'REC UNREAD', 'STO SENT', 'STO UNSENT', 'ALL'. Mensajes [leídos]/[no leídos], [enviados]/[no enviados], todos.
<b>AT+CMGD=&lt;index&gt;</b>	Elimina mensaje SMS.
<b>AT+CMGR=&lt;index&gt;</b>	Lee mensaje SMS específico.
<b>ATD+505#####;</b>	Realiza una llamada telefónica a determinado número.
<b>ATA</b>	Acepta llamada entrante.
<b>ATH</b>	Cuelga llamada entrante o saliente.

El SIM900 puede mantener una o múltiples conexiones. Una conexión GPRS única pasa por 9 estados, el MCU responsable de controlar el SIM900 debe emitir los comandos AT necesarios para permitir la conexión GPRS además de comprobar gradualmente el estado actual de dicha conexión. Luego de comprobar el estado, emitir el comando AT necesario para retomar la conexión. [46]



**Figura 2.9** Diagrama de estados para conexión GPRS única. [46]

Para conexión única los estados son 9 en total, después de ejecutar los respectivos comandos los estados deben ir cambiando, a como se aprecia en la figura 2.9.

## 2.3 Subsistema Microcontrolador

### PIC24FJ64GA002/dsPIC33FJ64GP202

Entre los tipos de procesadores elegimos un microcontrolador porque estos simplifican el circuito (Hardware) y reducen costos. Al seleccionar el microcontrolador debemos también tener en cuenta la disponibilidad de herramientas (Software/Hardware). Las compañías fabricantes de microcontroladores más accesibles y populares son: ATMEL, MICROCHIP, STMicroelectronics, NXP, TEXAS INSTRUMENTS. Algunos de los dispositivos evaluados pueden verse en la tabla 2.9.

Para nuestro proyecto básicamente necesitamos un microcontrolador con 2 UART, 1 SPI, con memoria flash superior a 32Kbytes y RAM 1K. Por razones de disponibilidad, costos (herramientas), seleccionamos de microchip de las familias PIC24 y dsPIC33F los modelos PIC24FJ64GA002 o dsPIC33FJ64GP202.

**Tabla 2.9** Lista de microcontroladores evaluados.

	<b>ATMEL</b>	<b>Microchip</b>	<b>STMicroelectronics</b>
<b>MCU</b>	megaAVR, AVR XMEGA	PIC24F, dsPIC33F	STM32
Núcleo	8/16 Bits	16 Bits	32 Bits ARM Cortex
Modelos	ATMEGA1284P ATMEGA328P ATMEGA644	PIC24FJ64GA002, dsPIC33FJ64GP202	STM32F103CB STM32F407VGT6 STM32F415RG

Elegimos de MICROCHIP los dispositivos: PIC24FJ64GA002 y dsPIC33FJ64GP202, las características de estos aparatos permiten que nuestro circuito sea compatible con otros modelos similares dentro de ambas familias. En estas familias hay gran variedad de memoria FLASH/RAM (128K/16K respectivamente), alta densidad de periféricos y pines mapeables. [47]

Estos microcontroladores de 16 Bits son muy flexibles lo que permite una fácil integración en proyectos ya que disponen de bastante memoria de programa/datos, encapsulado DIP, 28 pines de los cuales 16 son mapeables, 2 UART, 2 SPI, 1 I2C, etc.[48][49]

Podemos deducir que necesitamos como mínimo un procesador (Microcontrolador) con 2 interfaces seriales (UART), una interfaz SPI, al menos unos 8 pines I/O libres (siendo preferible que sus pines sean 3.3V TTL) y suficiente memoria FLASH/RAM para soportar nuestra aplicación. Los dispositivos que hemos seleccionados son el SIM900, L16/L76, microcontrolador PIC24FJ64GA002/dsPIC33FJ64GP202, tarjeta SD/MicroSD menor o igual a 32GB.

En la figura 2.10 se observa el diagrama de pin para el microcontrolador PIC24FJXXGA002 con sus pines numerados y con algunos periféricos no mapeables asignados a cada pin, se puede ver que hay pines nombrados desde la RP0 hasta RP15 estos son los pines mapeables a los cual el usuario puede asignar Timer, comparador, salida PWM, UART, SPI.

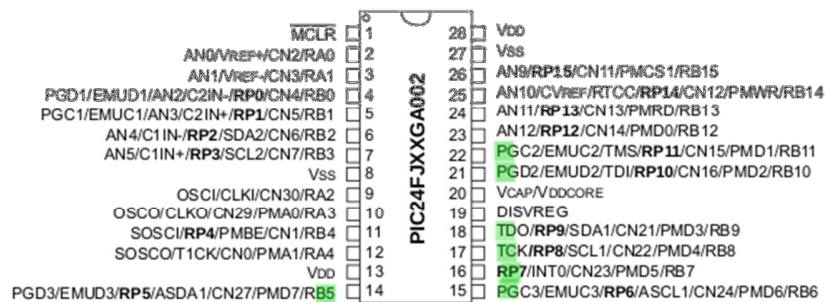


Figura 2.10 Microcontrolador PIC24FJxxGA002. [49]

## 2.4 Subsistema Almacenamiento (SDC)

Compuesto generalmente por la SDC (microSD o SD), es aquí donde respaldamos los datos localmente. Para que los datos sean fácilmente manipulados por el usuario con una computadora en archivos .txt o .csv, es necesario formatear la unidad con algún sistema de archivos, en nuestro caso FAT.

### Librerías SDC y Sistema de archivos FAT

Todas las tarjetas de memoria SD y SDIO (Input/Output) soportan el antiguo modo SPI/MMC que usa la interfaz serie de cuatro cables ligeramente más lenta (reloj, entrada serial, salida serial y selección de chip) que es compatible con los puertos SPI en muchos microcontroladores, la conexión puede apreciarse en la figura 2.11.

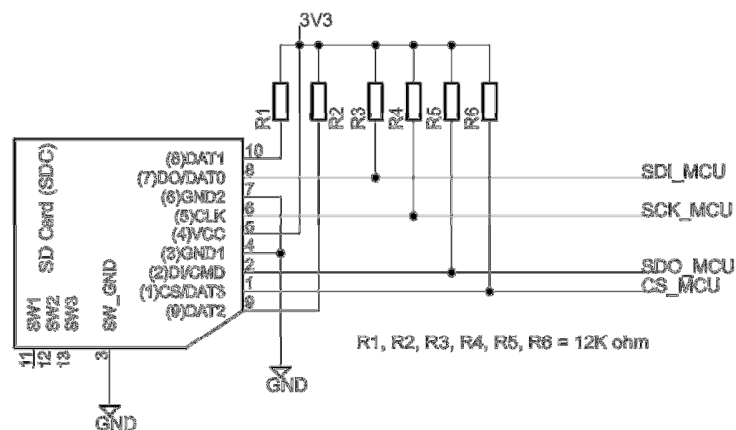
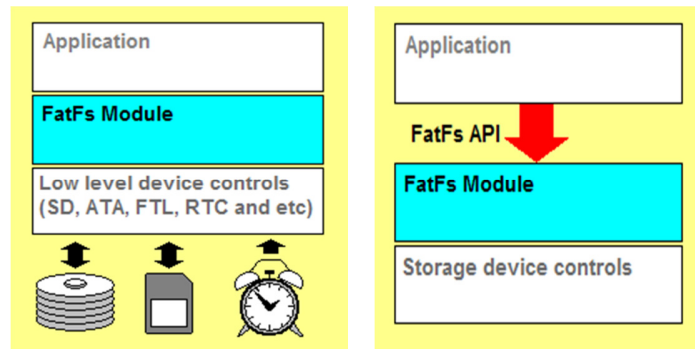


Figura 2.11 Circuito básico interfaz SPI – SDC.



**Figura 2.12** Modelo de Capas para FatFs. [51]

Para almacenar archivos en una SDC que puedan ser leídos en una computadora con Windows, Linux, etc. Es necesario el uso de un sistema de archivos estándar en este caso FAT (file allocation table) (12/16/32). [50] Para nuestro proyecto hemos seleccionado la librería FatFs **(by Elm-Chan)** la cual es una librería (open source) flexible, robusta y muy popular.

FatFs es un módulo software (lenguaje C) sistema de archivos FAT genérico para pequeños sistemas embebidos. El módulo FatFs es software libre, abierto para educación, investigación y desarrollo. Este puede ser modificado/redistribuido para proyectos personales o comerciales sin restricción alguna [51]. El módulo FatFs está escrito compatiblemente con ANSI C (C89) y completamente separado de la capa del dispositivo I/O como puede apreciarse en la figura 2.12. Es decir este es independiente de la plataforma y puede ser incorporado en pequeños microcontroladores con recursos limitados, como; 8051, PIC, AVR, Z80, etc. [51]

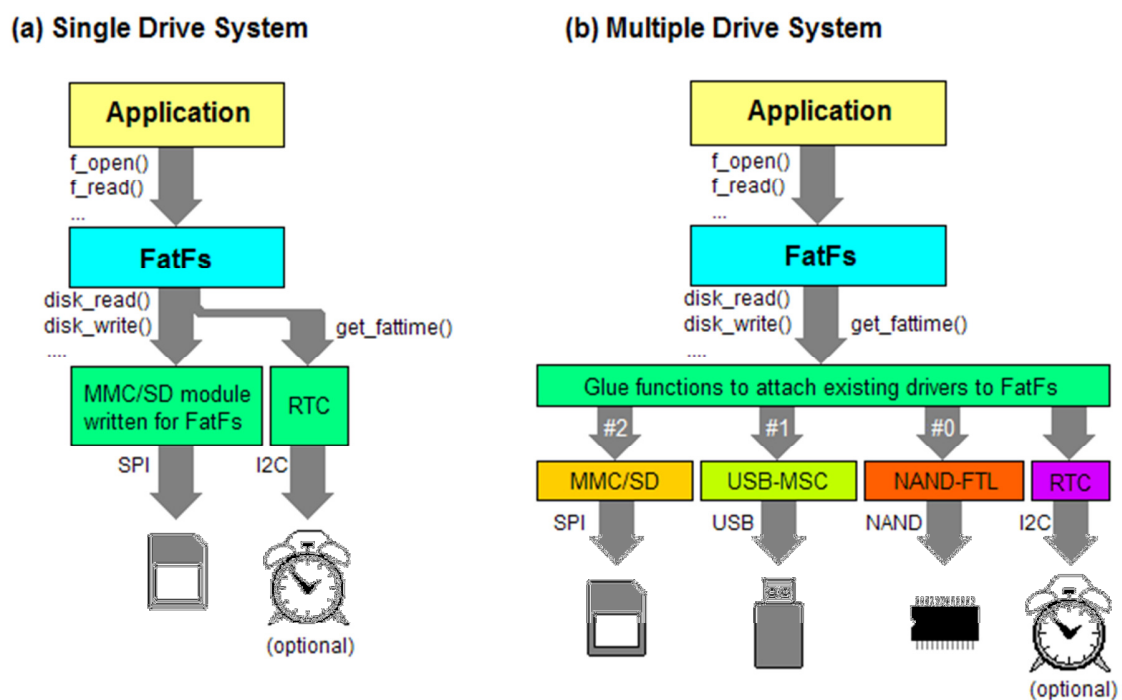
#### Especificaciones

- Sistema de archivos FAT compatible con Windows.
- Plataforma independiente.
- Poco uso de memoria de código y datos.
- Varias opciones de configuración:
  - Múltiple volumen (Discos Físicos y particiones).
  - Soporte para nombres largos ANSI/OEM, UNICODE.
  - Soporte RTOS para operación multi-tarea.



### Interfaz de control FatFs.

El módulo FatFs es una capa de sistema de archivos la cual es completamente independiente de la unidad física, como; tarjetas de memoria (SD/MicroSD/MMC), Discos Duros, USB y cualquier tipo de unidad de almacenamiento. FatFs obtiene acceso al dispositivo de almacenamiento vía una simple interfaz (código). El módulo de control en bajo nivel no es parte del módulo FatFs, este debe ser proporcionado por el implementador [51]. En la figura 2.13 se observan la conexión con distintos dispositivos de almacenamiento.



**Figura 2.13** Manejo de sistema Único/Múltiple. [51]

En la tabla 2.10 se pueden ver las principales funciones de la librería FatFs usadas en nuestro sistema con su descripción.

**Tabla 2.10** Principales funciones de la librería FatFs

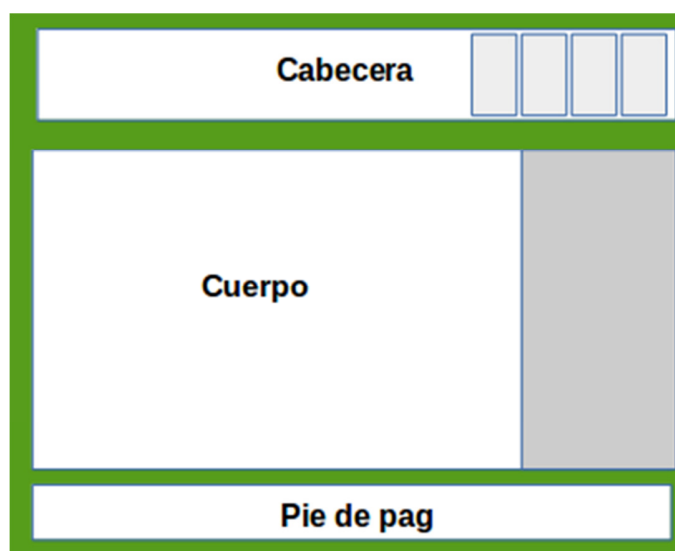
<b>f_mount</b>	Esta función registra/(desmonta) un sistema de archivos [objeto] al módulo FatFs. Es la primera función que debe ser llamada para montar la SDC.
<b>f_open</b>	Crea/abre un archivo objeto, con esta función se define el tipo de acceso al archivo; lectura/escritura, abrir si existe, abrir exista o no, crear si no existe, crear exista o no.  Ejemplo: <code>res = f_open (&amp;fil,"science.txt",FA_WRITE);</code>
<b>f_write</b>	Escribe dato al archivo en uso.
<b>f_read</b>	Lee datos de archivos en uso.
<b>f_lseek</b>	Mueve el puntero de lectura/escritura de un archivo abierto [objeto]. También puede ser usado para expandir la talla del archivo.
<b>f_close</b>	Cierra un archivo abierto. Si algún dato ha sido escrito al archivo, la información depositada del archivo es escrita al volumen. Después de que esta función es exitosamente ejecutada, el archivo objeto ya no es válido y puede ser descartado.  Nota: Si escribimos por ejemplo 1000 caracteres a un archivo y en un momento dado el sistema cayó antes de que pudiéramos ejecutar <b>f_close</b> , al abrir ese archivo veríamos que los 1000 caracteres no están. Es algo así como “guardar al salir” y como el sistema por error cayó y no salió correctamente se perdieron todos los datos escritos.
<b>f_sync</b>	Empuja la información depositada de un archivo en escritura. Esta función es como f_close (pero sin cerrar el archivo). Algo así como “guardar” lo escrito.  Es recomendable ejecutar esta función inmediatamente después de escribir (f_write) datos a la SD. Esto garantizará que ningún dato se pierda o al menos la menor pérdida de datos posible en caso de ocurrir un error.

## 2.5 Subsistema Visualización

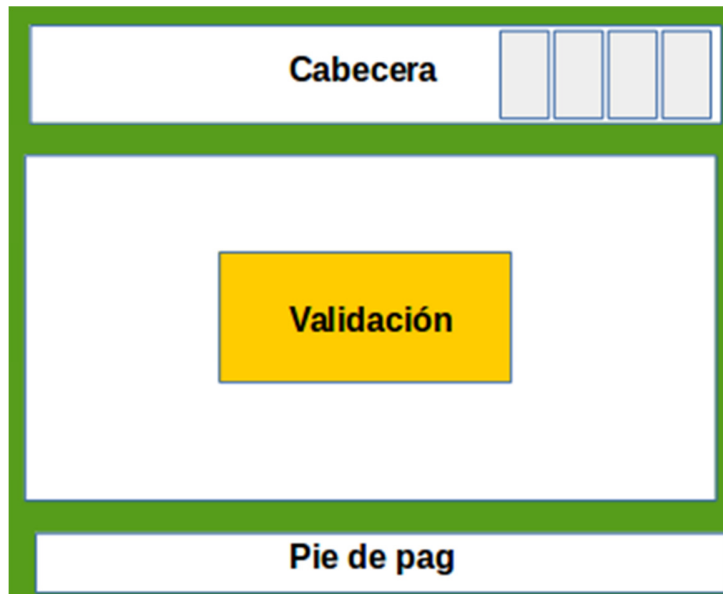
Como interfaz de visualización usamos una página web en don presentaremos los datos de geo localización referente al dispositivo AVL. La interfaz web consiste en una página con un formulario de validación desde donde insertaremos nuestro **usuario** y **password** , una vez el usuario es validado se presenta el último dato recibido del dispositivo AVL en Gmaps.

Para nuestra interfaz de visualización necesitamos un sitio de hosting web que albergue nuestra página, por razones económicas y disponibilidad hemos seleccionado hostinger.es el cual ofrece PHP y base de datos MySQL.

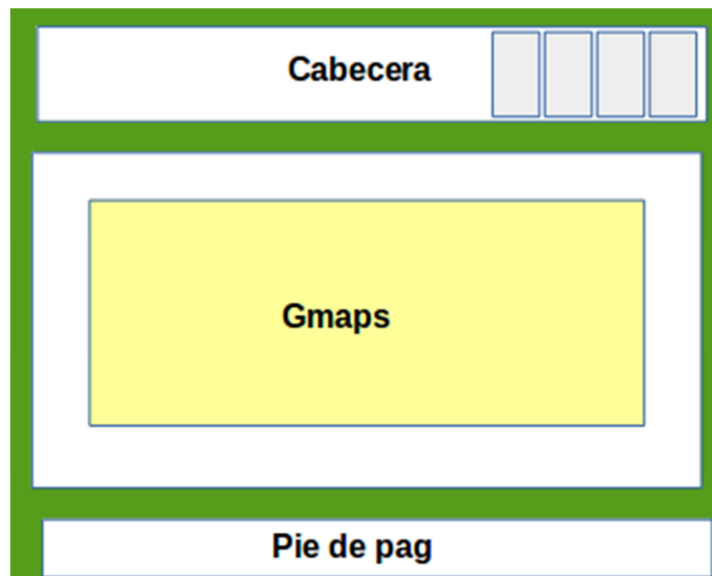
El propósito de la página web es presentar los datos de geo localización para el dispositivo AVL en tiempo real, en las figuras 2.14, 2.15 y 2.16 se aprecia el diseño básico de la página.



**Figura 2.14** Diseño página principal.



**Figura 2.15** Diseño validación usuario.



**Figura 2.16** Diseño vista tracking.

## **Descripción de Sitio web**

Para acceder al servicio de rastreo se nos pide autenticarnos (usuario, contraseña), si la autenticación es correcta se almacena en nuestro navegador la cookie necesaria para mantener la sesión activa. Una vez validado el usuario (con ayuda de PHP), como resultado nos envía un Google map embebido en la página web HTML. El mapa tiene los controles de ZOOM (+,-) y los controles de tipo de mapa. Con el control de tipo de mapa podemos cambiar de la vista tradicional de googlemaps a una vista satelital más detallada.

En el mapa recibido una vez estamos correctamente autenticados podemos observar un marcador indicando la latitud y la longitud de dicho punto donde se encuentra el marcador. Este marcador corresponde al último dato que ha recibido el servidor de parte del dispositivo AVL, es decir es el último dato almacenado en la base de datos. Además el sitio nos permite ver la hora y fecha de adquisición del último dato mostrado. En este momento todo dato enviado por el dispositivo AVL al servidor será automáticamente marcado en el mapa y podremos observar cómo se va marcando la trayectoria del dispositivo.

La actualización del mapa con el último dato en servidor se realiza mediante AJAX, con ayuda de este preguntamos frecuentemente al servidor por el último dato recibido del dispositivo, y si el dato es el mismo no lo marca más. En caso de que un nuevo dato ha sido recibido por el servidor pero su valor corresponde al mismo punto marcado anteriormente entonces el sitio solo actualiza la fecha y hora del dato.

En el servidor debemos tener una base de datos en la que se almacenen todos los datos enviados por el dispositivo AVL, también en la base de datos se debe guardar la información necesaria para autenticar un usuario (nombre de usuario, contraseña), ya que sin esto cualquiera podría ver el tracking, también en la tabla se necesita un campo en donde guardar el nombre completo del usuario.

En MySQL nuestras tablas son;

```
CREATE TABLE db.user (  
  userId INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  userKey VARCHAR(20) NOT NULL,  
  username VARCHAR(32) NOT NULL,  
  password_sha1 VARCHAR(32) NOT NULL,  
  tname VARCHAR(40) NOT NULL  
)default character set utf8;
```

```
CREATE TABLE db.device (  
  devId INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  devDate DATE NOT NULL,  
  devTime TIME NOT NULL,  
  devLat VARCHAR(32) NOT NULL,  
  devLon VARCHAR(32) NOT NULL  
)default character set utf8;
```

La tabla user almacena la información necesaria para que el usuario pueda acceder a su cuenta, es decir login y password además de una llave o key relacionada al dispositivo AVL, esta tabla incluye un campo para almacenar el nombre completo del usuario. En la tabla device se almacena la información referente al tracking del dispositivo AVL es decir fecha, hora de adquisición, latitud, longitud.

## 2.6 Integración (Prototipo). Desarrollo y evaluación por etapas.

Antes de poder integrar todos los elementos del sistema, fue necesario evaluar y someter a prueba cada etapa o subsistema que integran nuestro proyecto, en este punto realizamos pruebas independientes del hardware/software relacionado con GNSS, SDC, GSM/GPRS, Sitio WEB, etc.

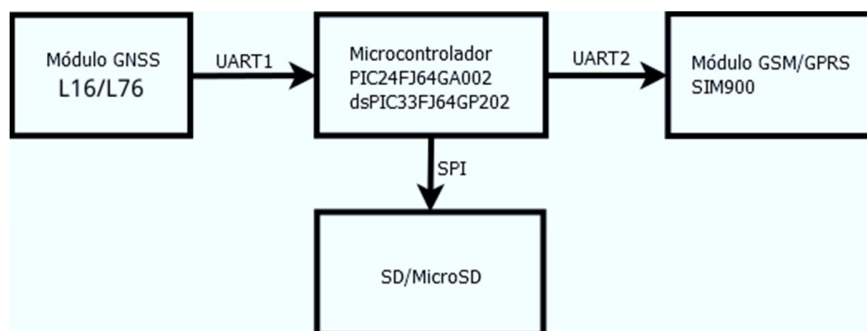


Figura 2.17 Bloques del dispositivo AVL.

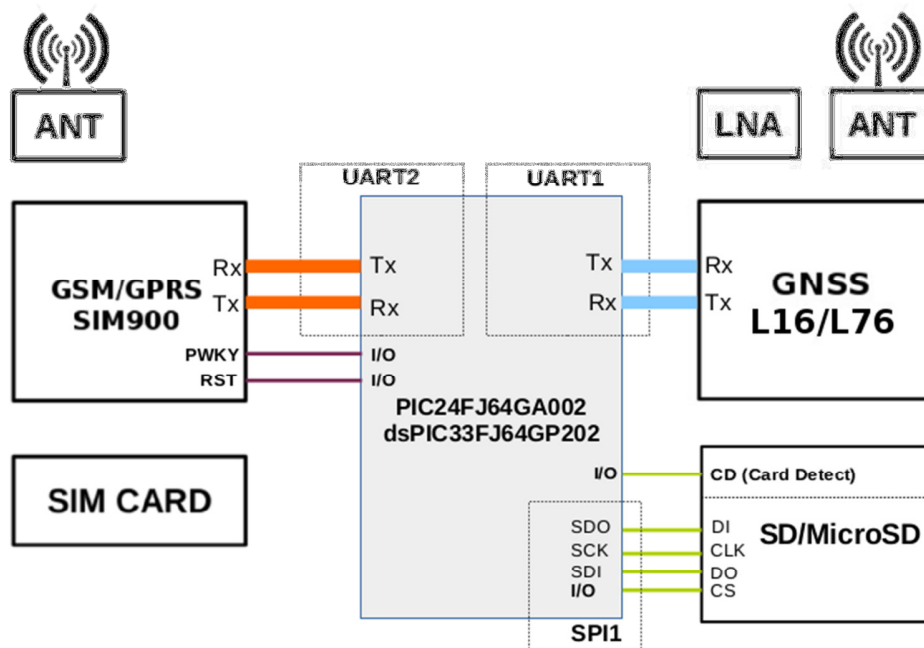


Figura 2.18 Conexiones, descripción general.

Las diferentes Etapas del dispositivo AVL con el modelo de dispositivo asociado a cada etapa pueden ser apreciadas en la figura 2.17, una descripción más detallada especificando las conexiones entre módulos con el microcontrolador está especificada en la figura 2.18.

**Etapas GNSS:** Para fines de pruebas conectamos únicamente el módulo L16/L76 al UART1 del microcontrolador. Lo configuramos con tasa de transferencia de datos de 9600 baudios (la cual ya viene por default) y con ayuda de un LCD podemos observar si los resultados mostrados son válidos. Hay que considerar que el módulo envía frecuentemente tramas de varias decenas de caracteres, nuestro microcontrolador debe ser capaz de recibir/procesar todos esos datos que entran por el UART1 en tiempo real, sin desatender las otras tareas. La manera óptima de manejar el módulo GNSS es mediante el uso de la interrupción de recepción del UART1, esta interrupción se activa cada vez que un carácter es recibido por el UART1.

Nuestra aplicación para el control del módulo L16/L76 podríamos decir; se hace en segundo plano con ayuda de la interrupción de recepción del UART1. Esto significa que en nuestro lazo principal de programa en (main) estará orientado al manejo de otras tareas.

Cada carácter recibido por el UART1 activa la interrupción y comprobamos si el dato recibido es un signo '\$' el cuál es el carácter de inicio de las tramas NMEA, en caso que no sea ese su valor será descartado. (Para que los datos recibidos por el UART1 sean tomados en cuenta como NMEA deben iniciar con '\$').

En caso de recibir un '\$' empezamos a capturar y procesar por partes la trama entrante (con ayuda del delimitador de campo el cual en las tramas NMEA son comas), esperamos a que toda la trama haya sido recibida esperando el carácter '\*' y los dos caracteres posteriores correspondientes al checksum. Una vez recibido el checksum validamos o invalidamos la trama. Con la llegada del primer delimitador el microcontrolador ya sabe qué tipo de trama está recibiendo y si por algún motivo la trama no corresponde a una conocida o es una trama no usada, sale inmediatamente del modo de captura de trama y pasa a esperar otra trama, es decir solo procesamos las tramas que vamos a utilizar las cuales para nosotros son RMC y GGA.

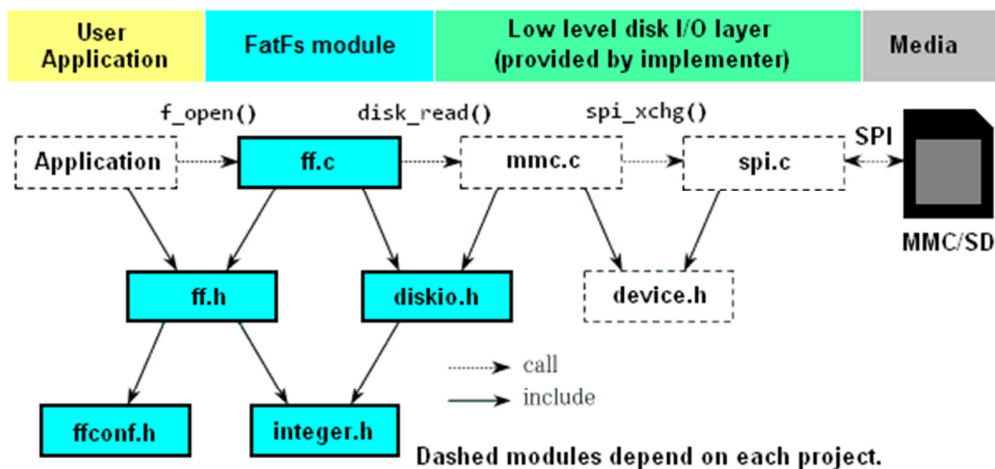


También realizamos pruebas GNSS utilizando únicamente el L10 QUECTEL/PMB-688 conjuntamente con la SD card. Extraemos los datos NMEA necesarios de la trama y los almacenamos en la SDC periódicamente cada par de segundos. Esta prueba es más adecuada que la anterior, porque tanto el dispositivo GPS como la SDC estarán integrados en nuestro dispositivo final además que nos permite de manera sencilla comprobar errores en el software/hardware ya sea para el módulo GNSS o la SDC.

Consideraciones importantes: Al usar interrupciones la tarea desarrollada dentro de dicha interrupción debe ser lo más ligeramente posible más aun cuando la interrupción se dispara con mucha frecuencia, debemos tener en cuenta que no genere solapamiento de datos o sobrecarga, por tal razón, la interrupción de recepción se encarga únicamente de recibir/descartar, revisar forma y almacenar, lo que no implica una carga excesiva para nuestro microcontrolador que tiene un rendimiento de hasta 40 MIPS para el dsPIC y 16 MIPS para el PIC24 ambos con núcleo de 16 bits.

Al momento de hacer las pruebas para el análisis de tramas NMEA debemos considerar la configuración actual del dispositivo GNSS en lo que se refiere a los sistemas GNSS activados en el módulo GNSS, esto es importante ya que el ID de mensaje para varios campos cambiara según el caso. Por ejemplo; al usar solo GPS el ID del mensaje para RMC es GPRMC, en caso de usar GPS + GLONASS el ID es GNRMC y en caso de usar GPS + BeiDou es BDRMC.

**Etapas SD:** En esta etapa para fines de pruebas conectamos únicamente la SD al SPI1 del microcontrolador, revisando si funciona de manera esperada. Como anteriormente habíamos dicho, utilizamos las librerías FatFs (byElm-Chan) para el control/manejo de la unidad de almacenamiento SD. Puede apreciarse en la figura 2.19 que para el uso de la librería se modela la implementación en 4 capas; Aplicación, FatFS, capa de bajo nivel, medio.



**Figura 2.19** Dependencia de archivos en las Capas FatFs.<sup>11</sup>

Para la SD lo que hay que proporcionar/programar es la **capa de bajo nivel** para acceder al medio (SD/MicroSD), teniendo en cuenta también las características del microcontrolador usado. Pero nosotros no tuvimos la necesidad de programar esta capa ya que en los ejemplos proporcionado por el propietario de la librería, venía incluida una para algunos modelos de PIC24 y dsPIC, la cual únicamente modificamos y adaptamos según nuestra necesidad.

En esta parte lo principal es verificar y comprobar las funciones, la manera de usarlas correctamente revisando los archivos generados e identificando posible fuente de errores. En nuestro dispositivo final el almacenaje de datos se hace periódicamente cada vez que hay datos validos a ser transmitidos por GPRS, en ese caso procede a almacenarse en disco, ya sea en forma binaria, ASCII, etc. El almacenaje de datos en nuestro caso se hace cada 20 segundos, éste es el tiempo en que enviamos un nuevo dato al servidor. Por lo tanto debemos comprobar los tiempos que demora en realizar distintas tareas y familiarizarnos con las funciones de la librería; crear archivos (.txt, .csv), realizar tareas de lectura/escritura, abrir/cerrar, guardar, etc. Y comprobar los tiempos.

<sup>11</sup> [http://elm-chan.org/fsw/ff/00index\\_e.html](http://elm-chan.org/fsw/ff/00index_e.html)

**Etapla GSM/GPRS:** Para fines de pruebas conectamos el SIM900 a una PC con ayuda de un cable SERIAL-USB, mediante un programa terminal verificamos los comandos y la forma de datos a ser insertados, aquí verificamos hacer un GET a un sitio web cualquiera, un GET exitoso con su determinado código de respuesta nos dice que estamos teniendo progresos y podemos pasar a desarrollar algoritmos y código necesario para controlar el SIM900.

Una vez comprobado con ayuda de la PC el correcto funcionamiento del módulo, comandos, respuestas y conexión GPRS. Desarrollamos los algoritmos, secuencias de comandos y código necesario que nos permita automatizar la tarea que envía datos a un servidor mediante GPRS.(En esta parte se debe tener operativo el sitio web al que vamos enviar los datos).

Pasamos a conectar únicamente el SIM900 al UART2 del microcontrolador, revisando si funciona de manera esperada. Todos los datos recibidos del SIM900 por el UART2 son almacenados con ayuda de la interrupción de recepción en un buffer desde el cual pueden ser analizados siempre que se necesiten, esto facilita determinar si un mensaje SMS ha sido recibido por el SIM900 pero hay que tener en cuenta que con cada revisión de buffer el puntero del buffer apunta al inicio.

Al enviarle un comando AT al SIM900 éste tiene que responder para que pueda recibir otro comando, es decir si aún está procesando un comando AT no puede recibir más y cualquier comando enviado al módulo no tiene validez. Por esta razón en nuestro código cuando enviamos un comando AT al SIM900 esperamos un tiempo determinado cada respuesta, en caso de no haber respuesta revisa el STATUS actual de la conexión, en caso no haber más respuestas, se reinicia el módulo por hardware con ayuda del pin RESET.

**Etapla Sitio WEB** Para este proyecto hemos desarrollado un pequeño sitio web que nos permita monitorear nuestro dispositivo AVL con ayuda de los mapas de Google. Para el desarrollo de ésta, contratamos un paquete de alojamiento web que proporcione base de dato (MySQL) aunque lo ideal sería contratar VPS (virtual private server) u otro servicio que nos proporcione mayor libertad para la administración del hardware.

Primeramente descargamos en Internet un diseño/estilo web gratuito los cuales están compuestos principalmente de archivos HTML/CSS. Modificamos dichos archivos para

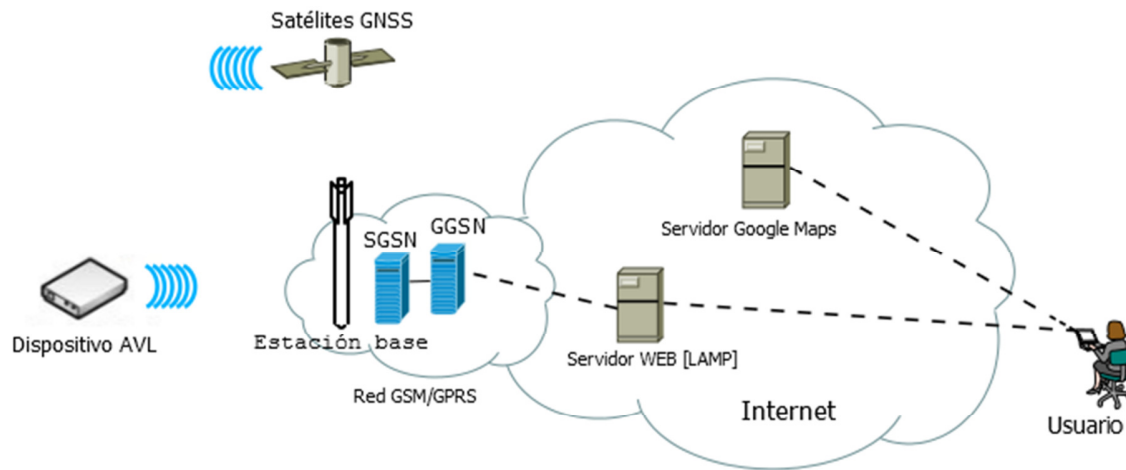
personalizar el sitio web, posteriormente adherimos el código PHP, JavaScript/AJAX, esto es lo que nos permite interactuar con el sitio y proporciona el servicio de monitoreo mediante mapas.

PHP es el motor de nuestro sitio WEB. PHP se ejecuta del lado del servidor y realiza todas las consultas/tareas relacionadas con la base de datos MySQL y proporciona el código HTML resultante de la validación de usuario/contraseña. Cuando nuestro dispositivo AVL envía datos al servidor con un GET/POST, es PHP quien se encarga de autenticar el dispositivo AVL y posteriormente se encarga de insertar los datos en la base de datos (MySQL).

Cuando un usuario se conecta a nuestro sitio web para monitorear el dispositivo AVL, el servidor mediante PHP responde con código HTML/JavaScript/AJAX embebido, una vez el código es descargado por el navegador este empieza a ejecutarse del lado del cliente. El usuario recibe la página web con mapa de google embebido, JavaScript/AJAX empieza a hacer preguntas al servidor sobre el último dato insertado en la base de datos (comprueba si este ya ha sido marcado en el mapa), el servidor mediante PHP hace la consulta a la base datos y da una respuesta.

Una vez nuestro sitio está activo para simular la conexión con el dispositivo AVL basta hacer un GET al archivo que se encarga de recibir los datos del dispositivo AVL desde el navegador o algún sitio que proporcione análisis de peticiones HTTP.

## 2.6 Funcionamiento (final) general del Dispositivo AVL.



**Figura 2.20** Sistema AVL (Descripción general)

El Dispositivo AVL se encarga de recibir las señales GNSS y enviarlas al servidor mediante la red GSM/GPRS, en la figura 2.20 aparece una representación genérica del sistema, en donde dentro de la red GSM/GPRS básicamente interviene el SGSN y el GGSN, para luego conectarse a internet en donde aparece nuestro servidor y el servidor GMAPS.

Al encender el dispositivo AVL el microcontrolador inicia/configura sus periféricos (Timer, UART, SPI, pines I/O, etc). Los retardos los realizamos utilizando el temporizador TIMER1 del microcontrolador, con dicho temporizador podemos llevar la cuenta de cuantos milisegundos tardan en ejecutarse algunas tareas, por ejemplo nosotros enviamos datos por la red GPRS cada 20 segundos y las funciones para el SIM900 necesitan retardos de algunos segundos los cuales los implementamos con el TIMER1 dejando libre la unidad de procesamiento principal.

Luego configuramos los pines de I/O. Los módulos UART a 9600 bps, 8 bits sin paridad. El SPI lo configuramos en modo maestro. Una vez configurados e iniciados los periféricos podemos proceder a inicializar los módulos L16/L76, SIM900, SDC, etc.

Cuando los módulos GNSS, GSM/GPRS y SDC estén ya inicializados, el microcontrolador espera la recepción de tramas NMEA validas (datos correctos), esto se debe a que no tiene sentido almacenar/transmitir datos erróneos. En el instante en que obtenemos datos válidos, el

microcontrolador activa la conexión GPRS e intenta enviar los datos por la red GSM/GPRS, simultáneamente revisa constantemente el estado de la conexión. Si la conexión se cae entonces el microcontrolador ordena reiniciarla otra vez.

Para conectarse al servidor y posteriormente enviar el datos; A nivel de la capa de transporte se establece una conexión TCP por el puerto 80 el cual ya sabemos está destinado a HTTP. Emitimos el siguiente comando y esperamos respuesta positiva "CONNECT OK".

**AT+CIPSTART="TCP","www.openavl.esy.es",80**

Como anteriormente habíamos dicho; el puerto 80 usa protocolo HTTP por lo tanto debemos usar elementos de dicho protocolo para acceder por ese puerto. Una vez conectados con el servidor, hacemos un GET o POST a un archivo PHP el cual se encarga de realizar la inserción de los datos en la base de datos (MySQL) alojada en nuestro servidor. Para el SIM900 el método debe finalizar con el carácter SUB (0x1A) el cual le indica al SIM900 empezar a transmitir el dato. En la tabla 2.11 se puede apreciar la forma como hacer un GET/POST para enviar los datos al servidor.

**Tabla 2.11 GET/POST HTTP**

<pre>GET /scripts/device.php?key=rkxRsX57d3R&amp;date=2015-01-08&amp;time=06:07:08&amp;lat=11.9233&amp;lon=-86.1433HTTP/1.0&lt;CR&gt;&lt;LF&gt; Host:openavl.esy.es&lt;CR&gt;&lt;LF&gt; Connection:close&lt;CR&gt;&lt;LF&gt; User-Agent:openavlnic&lt;CR&gt;&lt;LF&gt; &lt;CR&gt;&lt;LF&gt;</pre>
<pre>POST/scripts/device.phpHTTP/1.0&lt;CR&gt;&lt;LF&gt; Host: openavl.esy.es&lt;CR&gt;&lt;LF&gt; User-Agent: openavlnic&lt;CR&gt;&lt;LF&gt; Connection: close&lt;CR&gt;&lt;LF&gt; &lt;CR&gt;&lt;LF&gt; key=rkxRsX57d3R&amp;date=2015-01-08&amp;time=06:07:08&amp;lat=11.9142&amp;lon=-86.1433</pre>

Con los métodos anteriores, si se está utilizando sistema operativo Windows además de terminal Putty/Tera-Term, ir a opciones y configurar el fin de línea con <CR><LF> en caso contrario al final de cada línea se debe emitir la secuencias de teclas siguientes: <ctrl+m><ctrl+j>.

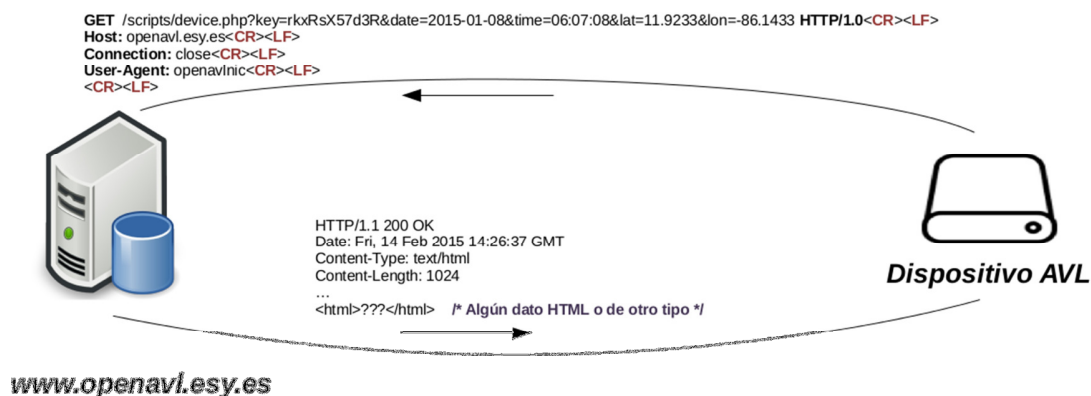
Nuestro dispositivo AVL usa el método GET para enviar los datos (No usamos el método POST). Podemos observar que hacemos la petición a un archivo llamado **device.php** ubicado en una carpeta llamada script en el directorio raíz. Los parámetros que entregamos a este archivo script están descritos en la tabla 2.12.

**Tabla 2.12** Campos enviados al servidor.

<b>key</b>	ID, Identificador del dispositivo	rxxRsX57d3
<b>date</b>	Fecha: yyyy-mm-dd	2015-01-08
<b>time</b>	Hora: hh:mm:ss	06:07:08
<b>lat</b>	Latitud: dd.dddd (grados)	11.9142
<b>lon</b>	Longitud: dd.dddd (grados)	-86.1433

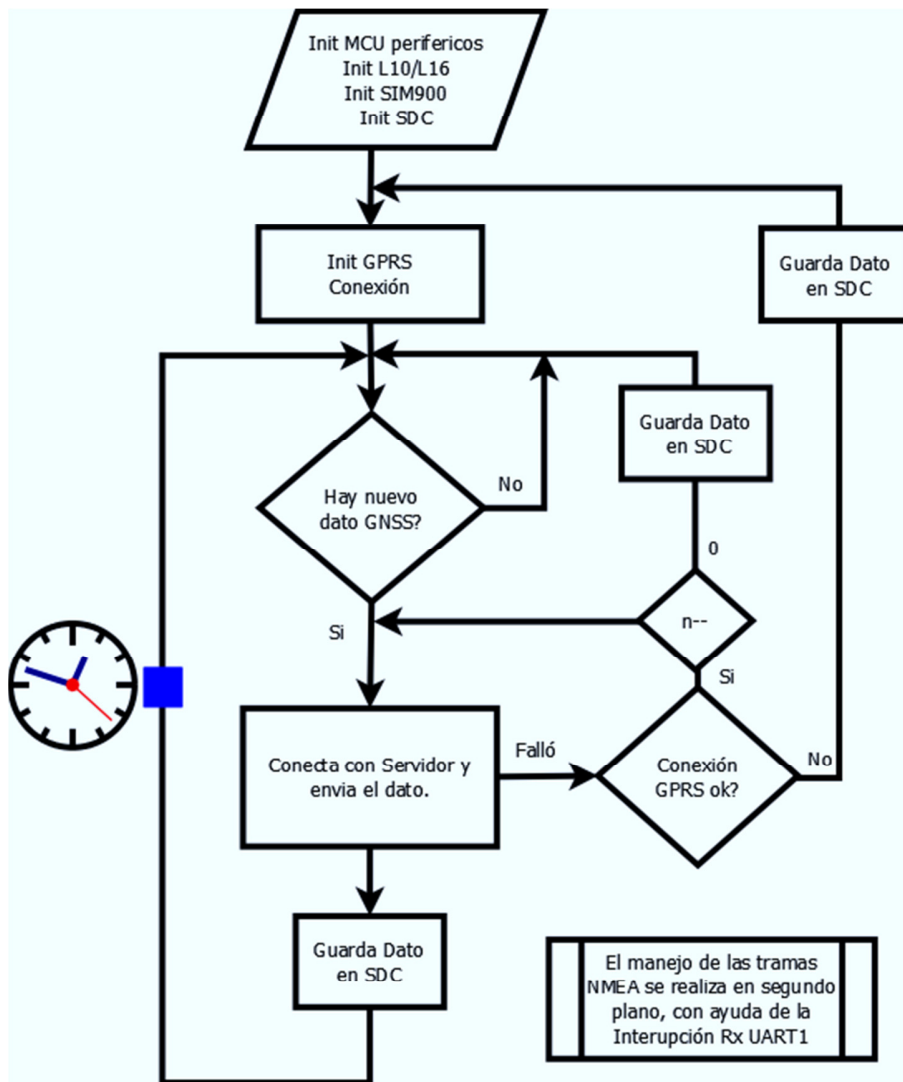
El HTTP del servidor responde con el código 200. Parecido a la siguiente respuesta:

<p><b>Possible Respuesta.</b></p> <p>HTTP/1.1 200 OK  Date: Fri, 14 Feb 2015 14:26:37 GMT  Content-Type: text/html  Content-Length: 1024  ...  &lt;html&gt;??&lt;/html&gt; /* <b>Algún dato HTML o de otro tipo</b> */</p>
--



**Figura 2.21** GET y respuesta al servidor.

Nuestro dispositivo AVL al enviar la petición GET espera por la respuesta “HTTP/1.1 200 OK” esto indica que nuestra petición fue correcta y se ejecutó sin problemas, dicha comunicación se puede apreciar en la figura 2.21. Además de un “AVL OK” enviado como respuesta por nuestro script, el cual indica que el ID (key) es válido y se han guardado los datos en la base de datos. En caso de que el ID (key) sea invalido no emite ningún código de respuesta

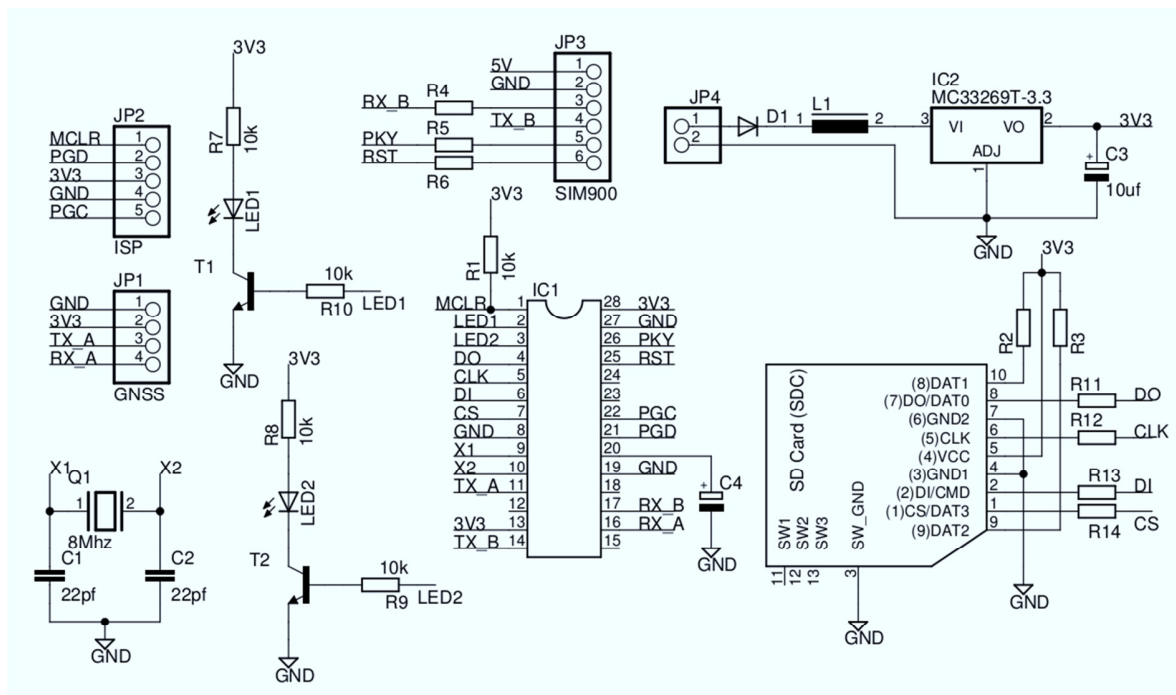


**Figura 2.22** Diagrama en bloques, operación del dispositivo AVL.



En la figura 2.22 se indica la secuencia de funcionamiento para el dispositivo AVL a modo de descripción general. El pequeño bloque azul es la condición de tiempo que le permite al dispositivo iniciar una nueva transmisión la cual para nuestro aparato es 20 segundos. También podemos ver el bloque rojo al cual se llega si el envío falló pero la conexión GPRS está activa, en este caso el microcontrolador ordena realizar otro intento de envío del dato. Si vuelve a fallar recolecta un nuevo dato y almacena en SDC el dato anterior.

En el circuito final el IC puede ser un PIC24 o un dsPIC33 con características similares por ejemplo **PIC24FJ64GA002**, **dsPIC33FJ64GP202**. El módulo GNSS puede ser un PMB-688, L10, L16, L26, L76 e igualmente el sistema trabajaría con normalidad pero lo referente al módulo GSM/GPRS estrictamente debe ser un SIM900 ya que los comandos AT tienen singularidades que varían según cada fabricante, el circuito final puede apreciarse en la figura 2.23.



**Figura 2.23** Circuito dispositivo AVL.

## Capítulo 3 Resultados

### Prueba Masatepe-Masaya (Ambiente Semi-Urbano)

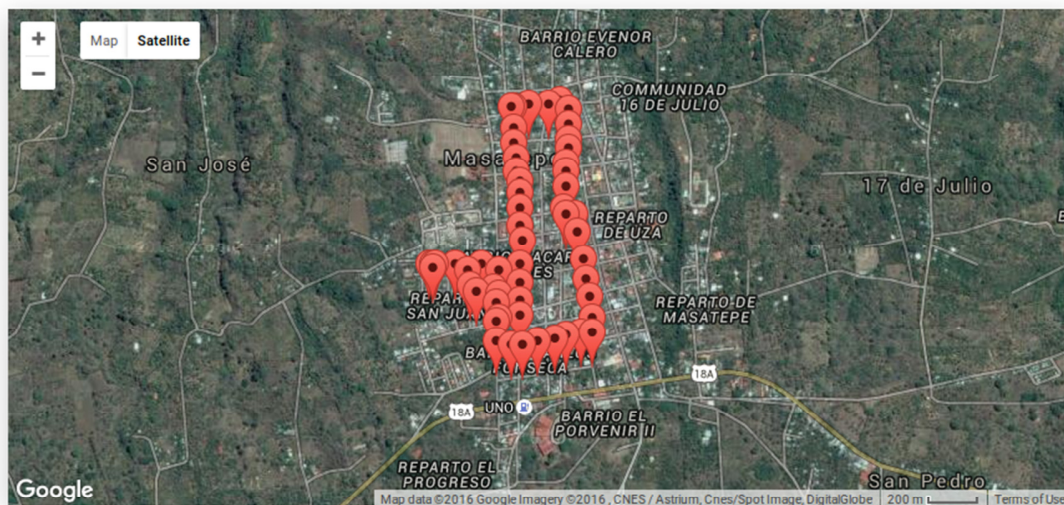
En la figura 3.1 se observa la trayectoria del dispositivo AVL que fue instalado en una motocicleta con la cual posteriormente se realizó un recorrido el 02/06/16, iniciando a las 23:35:13 UTC (5:35 PM) finalizando a las 23:53:41 UTC (5:53 PM). Un total de 101 datos fueron enviados al sitio web, almacenados en archivo de texto en la SDC y en la base de datos (MySQL) del servidor. Esta prueba fue realizada en un ambiente semi-urbano correspondiente al pueblo de Masatepe.

### Proyecto NicAVL

Tracking: Marco Munguía Mena

marcomunguia / Logout

Google Maps Update with AJAX



Last Data			
Date	Time	Latitud	Longitud
2016-06-02	23:53:41	11.9124	-86.1495

**Figura 3.1** Tracking de moto en la ciudad de Masatepe.

### Prueba Managua (Ambiente Urbano)

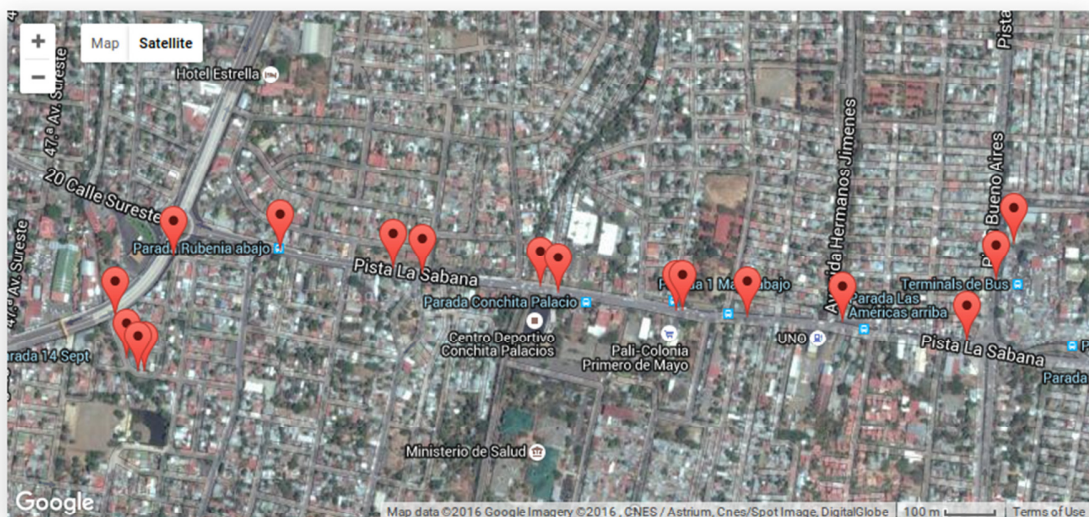
En la figura 3.2 se observa la trayectoria del dispositivo AVL instalado en un automóvil particular en el cual posteriormente se realizó un recorrido el 11/06/16, iniciando a las 16:56:13 UTC (10:56:13 AM) en la 14 de septiembre y finalizando a las 17:01:42 UTC (11:01 AM) en la parada de buses del Ivan. Un total de 17 datos fueron enviados al sitio web, almacenados en archivo de texto en la SDC y en la base de datos (MySQL) del servidor. Esta corta prueba fue realizada en un ambiente urbano correspondiente la ciudad de Managua.

## Proyecto NicAVL

Tracking: Marco Munguía

marcomunguia / Logout

Google Maps Update with AJAX

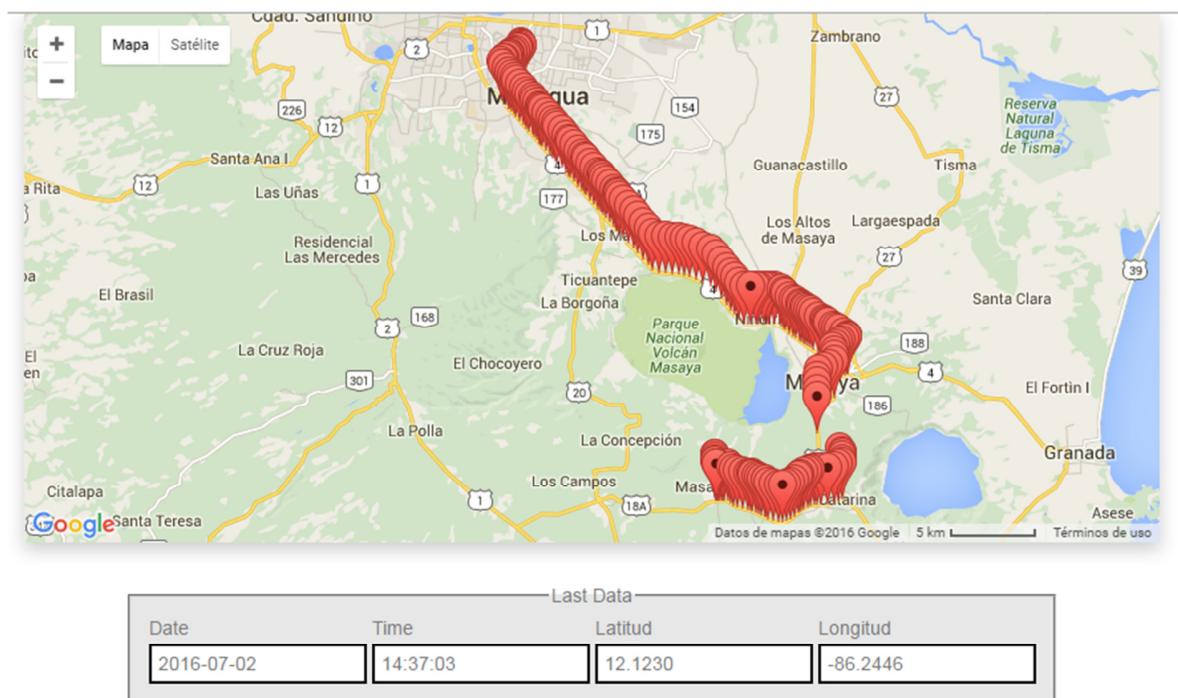


Last Data			
Date	Time	Latitud	Longitud
2016-06-11	17:01:42	12.1301	-86.2171

**Figura 3.2** Tracking de automóvil en la ciudad de Managua.

### Prueba desde Masatepe a Managua (Ambiente de Carretera).

En la figura 3.3 se observa la trayectoria del dispositivo AVL el que fue instalado en una unidad de transporte inter-local en la cual posteriormente se realizó un recorrido el 02/07/16, iniciando a las 13:21:13 UTC (7:21 AM) en el parque central de la ciudad de Masatepe y finalizando a las 14:37:03 UTC (8:37 AM) en el mercado Roberto Huembes de Managua. Un total de 219 datos fueron enviados al sitio web, almacenados en archivo de texto en la SDC y en la base de datos (MySQL) del servidor. Esta prueba fue realizada en un ambiente de carretera correspondiente al pueblo de Masatepe.



**Figura 3.3** Tracking de microbús ruta Masatepe-Managua.

## Capítulo 4 Conclusiones y recomendaciones

### 4.1 Conclusiones

En el presente documento hemos diseñado e implementado un dispositivo AVL usando las tecnologías GPS/GLONASS, GSM/GPRS, dicho dispositivo fue construido por etapas a fin de facilitar la rápida detección de posibles errores en el dispositivo final integrado, cada etapa del dispositivo fue sometida a pruebas individualmente.

Entre los resultados obtenidos podemos destacar varias pruebas realizadas en donde el sistema operó correctamente en ambientes semi-urbanos, urbanos y de carretera cuyo trayecto incluye partes urbanas y partes rurales de densa vegetación.

En la prueba de ambiente semi-urbano se rastreó de manera efectiva el dispositivo instalado en una motocicleta, dicha prueba fue realizada en la ciudad de Masatepe a una velocidad promedio menor de 25 kph. En la prueba de ambiente urbano se rastreó de manera efectiva el dispositivo instalado en un automóvil, dicha prueba fue realizada en la ciudad de Managua a una velocidad promedio de 60 kph. En la prueba de ambiente de carretera se rastreó de manera efectiva el dispositivo instalado en una unidad de transporte inter-urbano, esta prueba fue realizada desde Masatepe hasta Managua pasando por Catarina, Masaya a una velocidad promedio de 80 kph. El tiempo de envío entre lectura está configurado en 20 segundos aunque debemos tener en cuenta que depende de la disponibilidad de la red GPRS, apoyados en nuestras pruebas podemos ver que en su mayoría el lapso de tiempo entre transmisión se cumple en 20 segundos.



## 4.2 Recomendaciones

Recomendamos diseñar un PCB unificado ya que al hacer esto tendríamos aparte de un mejor uso de espacio físico la libertad de cambiar/seleccionar partes del dispositivo. Al diseñar un PCB unificado aconsejamos cambiar el microcontrolador PIC24/33 por un dispositivo ARM CORTEX-M3 (32 Bits) los cuales tienen un excelente rendimiento, bajo consumo y gran integración de periféricos.

Para este proyecto no usamos un dispositivo ARM porque estos no vienen en encapsulados PDIP, el encapsulado más fácil de manejar disponible de estos dispositivos es el LQFP el cual es un encapsulado de montaje superficial. De los microcontroladores ARM recomendamos usar dispositivos STM32F de **STmicroelectronics** la razón son las excelentes características en rendimiento, bajo consumo eléctrico, bajo costo de los dispositivos y las herramientas hardware/software proporcionadas por el fabricante para programar los dispositivos. También se puede usar dispositivos ATMEL ARM de 32 bits, NXP o de cualquier otro fabricante, esto permitiría implementar un mejor algoritmo y desarrollo del dispositivo.

Es necesario también el diseño de un sistema de alimentación (fuente) con cargador de batería para que el dispositivo AVL sea robusto y confiable, dicho diseño debería estar integrado en el PCB unificado. Mientras el dispositivo está en funcionamiento normal la batería de respaldo debe estar cargándose y en caso de una pérdida de la fuente principal la batería funcionaría como respaldo, también se deben agregar circuitos de protección.

Recomendamos el uso de actuadores para ejecutar tareas remotamente, es decir agregar código para procesar instrucciones recibidas ya sea mediante señales DTMF o código en SMS esto para activar relés, así el usuario podría remotamente suspender la inyección de combustible a un vehículo o activar una sirena en el mismo.

El dispositivo AVL almacena todas las tramas enviadas y las que no pudieron ser enviadas, cuando el dispositivo logra conectarse al servidor luego de intentos de transmisión fallidos, este no reenvía los datos almacenados en SD que no fueron exitosamente transmitidos. Por tal razón recomendamos crear una función que se encargue de agrupar y transmitir todos los datos que no pudieron ser enviados.

## VI. Bibliografía

- [1]. [https://en.wikipedia.org/wiki/Automatic\\_vehicle\\_location](https://en.wikipedia.org/wiki/Automatic_vehicle_location)
- [2]. J. Sanz Subirana, J.M. Juan Zornoza and M. Hernández-Pajares (2013). GNSS DATA PROCESSING, Volume I: Fundamentals and Algorithms (1ª ed). Netherlands. European Space Agency (ESA)
- [3]. Microchip 2010 Master Conference. (GPS) Aplicaciones GSM/GPS con MCU's MICROCHIP. Microchip Technology Incorporated.
- [4]. <http://www.opentracker.tigal.com>
- [5]. Feng, Y (2003), Combined Galileo and GPS: A Technical Perspective. Journal of Global Positioning Systems, 2 (1): 67-72.
- [6]. <http://www.asamblea.gob.ni/339776/convenio-de-cooperacion-para-actividades-espaciales-nicaragua-rusia/>
- [7]. <http://whatis.techtarget.com/definition/automatic-vehicle-locator-AVL>
- [8]. Michael Carpenter (2009) Interfacing a GPS receiver with a microprocessor via UART. Application Note.
- [9]. C. Tröster-Schmid (2014). GPS, GLONASS, GALILEO, BEIDOU Receiver testing using a GNSS signal simulator Application Note. Munich-Germany. Rohde & Schwarz GmbH & Co. KG
- [10]. Gregory T. French (1996). Understanding The GPS an introduction to the global positioning system (1ª ed). United States of America: GeoResearch.Inc.
- [11]. GuochangXu (2007). GPS Theory, Algorithms and Applications (2ª ed). UnitedStates of America: Springer.
- [12]. <http://www.gpsinformation.org>
- [13]. National Marine Electronics Association (2002). NMEA 0183 Standard For Interfacing Marine Electronic Devices (Version 3.0.4.).
- [14]. BERND EISSFELLER, GERALD AMERES, VICTORIA KROPP, DANIEL SANROMA (2007), Performance of GPS, GLONASS and Galileo. München-Germany.
- [15]. Charles Jeffrey (2010). An Introduction to GNSS. GPS, GLONASS, GALILEO and other navigation satellite systems (1ª ed). Canada. NovAtel Inc.
- [16]. Juan Andrés Sánchez Wevar (2005). Análisis y Estudio de redes GPRS. (Trabajo de Titulación). Universidad Austral de Chile.

- [17]. Jukka Lempiäinen, Matti Manninen (2002). RADIO INTERFACE SYSTEM PLANNING FOR GSM/GPRS/UMTS. United States of America. KluwerAcademicPublishers.
- [18]. Nokia Networks Oy (2004). GPRS Architecture: Interfaces and Protocols, Training Document GPRS System Course.
- [19]. Håkan Granbohm and Joakim Wiklund (1999). GPRS—General packet radio service (Review No. 2). Ericsson.
- [20]. [https://en.wikipedia.org/wiki/Hayes\\_command\\_set](https://en.wikipedia.org/wiki/Hayes_command_set)
- [21]. [https://en.wikibooks.org/wiki/Serial\\_Programming/Modems\\_and\\_AT\\_Commands](https://en.wikibooks.org/wiki/Serial_Programming/Modems_and_AT_Commands)
- [22]. SIMCOM (2010). SIM900 AT Command Manual. (version 1.03). Shanghai SIMCom Wireless Solutions Ltd.
- [23]. SIMCOM (2010). SIM900 AT Commands Set. (version 1.00). Shanghai SIMCom Wireless Solutions Ltd.
- [24]. Richard H. Barnett, Sarah Cox and Larry O’Cull (2007). Embedded C programming and the Atmel AVR (2<sup>a</sup> ed.). Canadá: Thompson Delmar.
- [25]. Microchip. MPLAB ® XC16 C Compiler User’s Guide. Microchip Technology Inc.
- [26]. MICROCHIP (2009). PICkit™ 3, Programmer/Debugger, User’s Guide. Microchip Technology Inc.
- [27]. Lucio Di Jasio (2007). Programming 16-Bit PIC Microcontrollers in C, Learning to Fly the PIC24. (1<sup>a</sup> ed.). UnitedStates of America. Elsevier Inc.
- [28]. Lynn Beighley and Michael Morrison (2009). Head First PHP & MySQL ( 1<sup>a</sup> ed.). UnitedStates of America. O’Reilly Media, Inc.
- [29]. Elisabeth and Eric Freeman (2005). Head First HTML with CSS and XHTML (1<sup>a</sup> ed.). UnitedStates of America. O’Reilly Media, Inc.
- [30]. Robin Nixon (2014). Learning PHP, MySQL, JavaScript, CSS & HTML5. ( 3<sup>a</sup> ed.). UnitedStates of America. O’Reilly Media, Inc.
- [31]. Ellie Quigley (2011). JavaScript by Example (2<sup>a</sup> ed.). United States of America. Prentice Hall.
- [32]. HERE (2014). Maps API for JavaScript Developer’s Guide (Version 3.0.4.). UnitedStates of America.
- [33]. Geospatial Training & Consulting, LLC (2006). Google Maps API; The New World of Web Mapping (Version 2). UnitedStates of America. Geospatial Training &Consulting, LLC.
- [34]. Schuyler Erle, Rich Gibson (2006). Google Maps Hacks (1<sup>a</sup> ed.). United States of America. O’Reilly Media, Inc.
- [35]. Kevin Yank (2009). Build Your Own Database Driven Web Site Using PHP & MySQL (4<sup>a</sup> ed.). UnitedStates of America. SitePointPty. Ltd.



- [36]. Lynn Beighley (2007). Head First SQL (1<sup>st</sup> ed.). United States of America. O'Reilly Media, Inc.
- [37]. Lynn Beighley and Michael Morrison (2009). Head First PHP & MySQL (1<sup>st</sup> ed.). United States of America. O'Reilly Media, Inc.
- [38]. Brett McLaughlin (2012). PHP and MySQL: The Missing Manual. (1<sup>st</sup> ed.). United States of America. O'Reilly Media, Inc.
- [39]. Phil Ballard , Michael Moncur (2009). Sams Teach Yourself Ajax, JavaScript, and PHP All in One. (1<sup>st</sup> ed.). United States of America. Sams Publishing.
- [40]. QUECTEL (2015-11-11). GNSS Module Series L76 Series Hardware Design. (version 2.1). QuectelWirelessSolution Co., Ltd.
- [41]. QUECTEL (2012). L16 Hardware Design. (version 1.0). Quectel Wireless Solution Co., Ltd.
- [42]. QUECTEL (2015-11-11). L76 Series Hardware Design. (version 2.1). Quectel Wireless Solution Co., Ltd.
- [43]. QUECTEL (2015-12-03). GNSS Module Series L76 Series Protocol Specification. (version 2.0). QuectelWirelessSolution Co., Ltd.
- [44]. TAOGLAS. AA.161.301111 Specification, GPS/GLONASS IP67 Active antenna. Taoglas Ltd.
- [45]. SIMCOM (2010). SIM900 Hardware Design. (version 2.00). Shanghai SIMCom Wireless Solutions Ltd.
- [46]. SIMCOM (2010). TCPIP Application Note. (version 1.01). Shanghai SIMCom Wireless Solutions Ltd.
- [47]. MICROCHIP (2005). dsPIC® Digital Signal Controllers , The Best of Both Worlds. Microchip Technology Inc.
- [48]. MICROCHIP (2009). dsPIC33FJ32GP302/304, dsPIC33FJ64GPX02/X04, and dsPIC33FJ128GPX02/X04 Data Sheet, High-Performance, 16-bit Digital Signal Controllers. Microchip Technology Inc.
- [49]. MICROCHIP (2010). PIC24FJ64GA004 Family Data Sheet, 28/44-Pin General Purpose, 16-Bit Flash Microcontrollers. Microchip Technology Inc.
- [50]. Martin Bates (2006). Interfacing PIC microcontrollers, Embedded design by interactive simulation (1<sup>st</sup> ed) United States of America: Elsevier Inc.
- [51]. FatFs Library. [www.elm-chan.org](http://www.elm-chan.org)

## VII. Anexos



### dsPIC33FJ32GP302/304, dsPIC33FJ64GPX02/X04, AND dsPIC33FJ128GPX02/X04

#### High-Performance, 16-Bit Digital Signal Controllers

##### Operating Range:

- Up to 40 MIPS operation (at 3.0-3.6V):
  - Industrial temperature range (-40°C to +85°C)
  - Extended temperature range (-40°C to +125°C)
- Up to 20 MIPS operation (at 3.0-3.6V):
  - High temperature range (-40°C to +140°C)

##### High-Performance DSC CPU:

- Modified Harvard architecture
- C compiler optimized instruction set
- 16-bit wide data path
- 24-bit wide instructions
- Linear program memory addressing up to 4M instruction words
- Linear data memory addressing up to 64 Kbytes
- 83 base instructions: mostly 1 word/1 cycle
- Two 40-bit accumulators with rounding and saturation options
- Flexible and powerful addressing modes:
  - Indirect
  - Modulo
  - Bit-Reversed
- Software stack
- 16 x 16 fractional/integer multiply operations
- 32/16 and 16/16 divide operations
- Single-cycle multiply and accumulate:
  - Accumulator write back for DSP operations
  - Dual data fetch
- Up to  $\pm 16$ -bit shifts for up to 40-bit data

##### Direct Memory Access (DMA):

- 8-channel hardware DMA
- Up to 2 Kbytes dual ported DMA buffer area (DMA RAM) to store data transferred via DMA:
  - Allows data transfer between RAM and a peripheral while CPU is executing code (no cycle stealing)
- Most peripherals support DMA

##### Timers/Capture/Compare/PWM:

- Timer/Counters, up to five 16-bit timers:
  - Can pair up to make two 32-bit timers
  - One timer runs as a Real-Time Clock with an external 32.768 kHz oscillator
  - Programmable prescaler
- Input Capture (up to four channels):
  - Capture on up, down or both edges
  - 16-bit capture input functions
  - 4-deep FIFO on each capture
- Output Compare (up to four channels):
  - Single or Dual 16-bit Compare mode
  - 16-bit Glitchless PWM mode
- Hardware Real-Time Clock/Calendar (RTCC):
  - Provides clock, calendar and alarm functions

##### Interrupt Controller:

- 5-cycle latency
- Up to 49 available interrupt sources
- Up to three external interrupts
- Seven programmable priority levels
- Five processor exceptions

##### Digital I/O:

- Peripheral pin Select functionality
- Up to 35 programmable digital I/O pins
- Wake-up/Interrupt-on-Change for up to 31 pins
- Output pins can drive from 3.0V to 3.6V
- Up to 5V output with open drain configuration
- All digital input pins are 5V tolerant
- 4 mA sink on all I/O pins

##### On-Chip Flash and SRAM:

- Flash program memory (up to 128 Kbytes)
- Data SRAM (up to 16 Kbytes)
- Boot, Secure and General Security for program Flash

## dsPIC33FJ32GP302/304, dsPIC33FJ64GPX02/X04, AND dsPIC33FJ128GPX02/X04

### System Management:

- Flexible clock options:
  - External, crystal, resonator, internal RC
  - Fully integrated Phase-Locked Loop (PLL)
  - Extremely low jitter PLL
- Power-up Timer
- Oscillator Start-up Timer/Stabilizer
- Watchdog Timer with its own RC oscillator
- Fail-Safe Clock Monitor
- Reset by multiple sources

### Power Management:

- On-chip 2.5V voltage regulator
- Switch between clock sources in real time
- Idle, Sleep, and Doze modes with fast wake-up

### Analog-to-Digital Converters (ADCs):

- 10-bit, 1.1 Msps or 12-bit, 500 ksps conversion:
  - Two and four simultaneous samples (10-bit ADC)
  - Up to 13 input channels with auto-scanning
  - Conversion start can be manual or synchronized with one of four trigger sources
  - Conversion possible in Sleep mode
  - $\pm 2$  LSB max integral nonlinearity
  - $\pm 1$  LSB max differential nonlinearity

### Audio Digital-to-Analog Converter (DAC):

- 16-bit Dual Channel DAC module
- 100 ksps maximum sampling rate
- Second-Order Digital Delta-Sigma Modulator

### Data Converter Interface (DCI) module:

- Codec interface
- Supports I<sup>2</sup>S and AC'97 protocols
- Up to 16-bit data words, up to 16 words per frame
- 4-word deep TX and RX buffers

### Comparator Module:

- Two analog comparators with programmable input/output configuration

### CMOS Flash Technology:

- Low-power, high-speed Flash technology
- Fully static design
- 3.3V ( $\pm 10\%$ ) operating voltage
- Industrial and Extended temperature
- Low power consumption

### Communication Modules:

- 4-wire SPI (up to two modules):
  - Framing supports I/O interface to simple codecs
  - Supports 8-bit and 16-bit data
  - Supports all serial clock formats and sampling modes
- I<sup>2</sup>C™:
  - Full Multi-Master Slave mode support
  - 7-bit and 10-bit addressing
  - Bus collision detection and arbitration
  - Integrated signal conditioning
  - Slave address masking
- UART (up to two modules):
  - Interrupt on address bit detect
  - Interrupt on UART error
  - Wake-up on Start bit from Sleep mode
  - 4-character TX and RX FIFO buffers
  - LIN bus support
  - IrDA® encoding and decoding in hardware
  - High-Speed Baud mode
  - Hardware Flow Control with CTS and RTS
- Enhanced CAN (ECAN™ module) 2.0B active:
  - Up to eight transmit and up to 32 receive buffers
  - 16 receive filters and three masks
  - Loopback, Listen Only and Listen All
  - Messages modes for diagnostics and bus monitoring
  - Wake-up on CAN message
  - Automatic processing of Remote Transmission Requests
  - FIFO mode using DMA
  - DeviceNet™ addressing support
- Parallel Master Slave Port (PMP/EPSP):
  - Supports 8-bit or 16-bit data
  - Supports 16 address lines
- Programmable Cyclic Redundancy Check (CRC):
  - Programmable bit length for the CRC generator polynomial (up to 16-bit length)
  - 8-deep, 16-bit or 16-deep, 8-bit FIFO for data input

### Packaging:

- 28-pin SDIP/SOIC/QFN-S
- 44-pin TQFP/QFN

**Note:** See the device variant tables for exact peripheral features per device.

## dsPIC33FJ32GP302/304, dsPIC33FJ64GPX02/X04, AND dsPIC33FJ128GPX02/X04

### dsPIC33FJ32GP302/304, dsPIC33FJ64GPX02/X04, AND dsPIC33FJ128GPX02/X04 PRODUCT FAMILIES

The device names, pin counts, memory sizes, and peripheral availability of each device are listed below. The following pages show their pinout diagrams.

#### dsPIC33FJ32GP302/304, dsPIC33FJ64GPX02/X04, and dsPIC33FJ128GPX02/X04 Controller Families

Device	Pins	Program Flash Memory (Kbyte)	RAM (Kbyte) <sup>(1)</sup>	Remappable Peripheral									RTCC	I <sup>2</sup> C™	CRC Generator	10-bit/12-bit ADC (Channels)	16-bit Audio DAC (Pins)	Analog Comparator (2 Channels/Voltage Regulator)	8-bit Parallel Master Port (Address Lines)	I/O Pins	Packages
				Remappable Pins	16-bit Timer <sup>(2)</sup>	Input Capture	Output Compare Standard PWM	Data Converter Interface	UART	SPI	ECAN™	External Interrupts <sup>(3)</sup>									
dsPIC33FJ128GP804	44	128	16	26	5	4	4	1	2	2	1	3	1	1	1	13	6	1/1	11	35	QFN TQFP
dsPIC33FJ128GP802	28	128	16	16	5	4	4	1	2	2	1	3	1	1	1	10	4	1/0	2	21	SDIP SOIC QFN-S
dsPIC33FJ128GP204	44	128	8	26	5	4	4	1	2	2	0	3	1	1	1	13	0	1/1	11	35	QFN TQFP
dsPIC33FJ128GP202	28	128	8	16	5	4	4	1	2	2	0	3	1	1	1	10	0	1/0	2	21	SDIP SOIC QFN-S
dsPIC33FJ64GP804	44	64	16	26	5	4	4	1	2	2	1	3	1	1	1	13	6	1/1	11	35	QFN TQFP
dsPIC33FJ64GP802	28	64	16	16	5	4	4	1	2	2	1	3	1	1	1	10	4	1/0	2	21	SDIP SOIC QFN-S
dsPIC33FJ64GP204	44	64	8	26	5	4	4	1	2	2	0	3	1	1	1	13	0	1/1	11	35	QFN TQFP
dsPIC33FJ64GP202	28	64	8	16	5	4	4	1	2	2	0	3	1	1	1	10	0	1/0	2	21	SDIP SOIC QFN-S
dsPIC33FJ32GP304	44	32	4	26	5	4	4	1	2	2	0	3	1	1	1	13	0	1/1	11	35	QFN TQFP
dsPIC33FJ32GP302	28	32	4	16	5	4	4	1	2	2	0	3	1	1	1	10	0	1/0	2	21	SDIP SOIC QFN-S

**Note** 1: RAM size is inclusive of 2 Kbytes of DMA RAM for all devices except dsPIC33FJ32GP302/304, which include 1 Kbyte of DMA RAM.  
2: Only four out of five timers are remappable.  
3: Only two out of three interrupts are remappable.

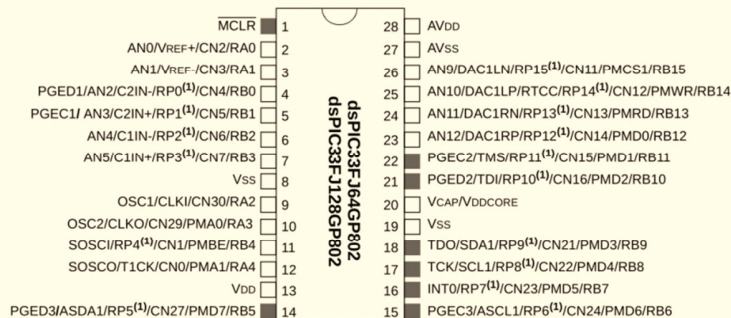


## dsPIC33FJ32GP302/304, dsPIC33FJ64GPX02/X04, AND dsPIC33FJ128GPX02/X04

### Pin Diagrams

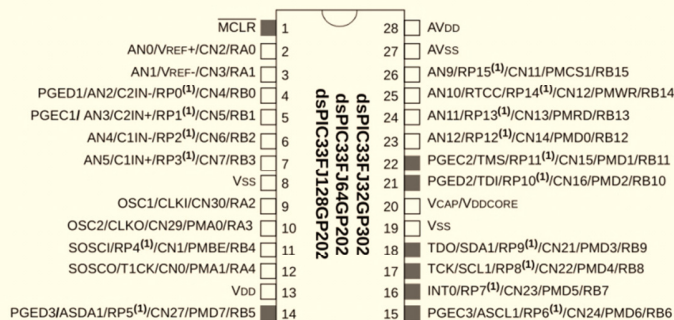
#### 28-Pin SDIP, SOIC

■ = Pins are up to 5V tolerant




#### 28-Pin SDIP, SOIC

■ = Pins are up to 5V tolerant



**Note 1:** The RPx pins can be used by any remappable peripheral. See the table "dsPIC33FJ32GP302/304, dsPIC33FJ64GPX02/X04, and dsPIC33FJ128GPX02/X04 Controller Families" in this section for the list of available peripherals.



# MICROCHIP PIC24FJ64GA004 FAMILY

## 28/44-Pin General Purpose, 16-Bit Flash Microcontrollers

### High-Performance CPU:

- Modified Harvard Architecture
- Up to 16 MIPS Operation @ 32 MHz
- 8 MHz Internal Oscillator with 4x PLL Option and Multiple Divide Options
- 17-Bit by 17-Bit Single-Cycle Hardware Multiplier
- 32-Bit by 16-Bit Hardware Divider
- 16-Bit x 16-Bit Working Register Array
- C Compiler Optimized Instruction Set Architecture:
  - 76 base instructions
  - Flexible addressing modes
- Two Address Generation Units for Separate Read and Write Addressing of Data Memory

### Special Microcontroller Features:

- Operating Voltage Range of 2.0V to 3.6V
- 5.5V Tolerant Input (digital pins only)
- High-Current Sink/Source (18 mA/18 mA) on All I/O Pins
- Flash Program Memory:
  - 10,000 erase/write
  - 20-year data retention minimum
- Power Management modes:
  - Sleep, Idle, Doze and Alternate Clock modes
  - Operating current 650  $\mu$ A/MIPS typical at 2.0V
  - Sleep current 150 nA typical at 2.0V
- Fail-Safe Clock Monitor Operation:
  - Detects clock failure and switches to on-chip, low-power RC oscillator
- On-Chip, 2.5V Regulator with Tracking mode
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Flexible Watchdog Timer (WDT) with On-Chip, Low-Power RC Oscillator for Reliable Operation
- In-Circuit Serial Programming™ (ICSP™) and In-Circuit Debug (ICD) via 2 Pins
- JTAG Boundary Scan Support

### Analog Features:

- 10-Bit, up to 13-Channel Analog-to-Digital Converter:
  - 500 ksp/s conversion rate
  - Conversion available during Sleep and Idle
- Dual Analog Comparators with Programmable Input/Output Configuration

### Peripheral Features:

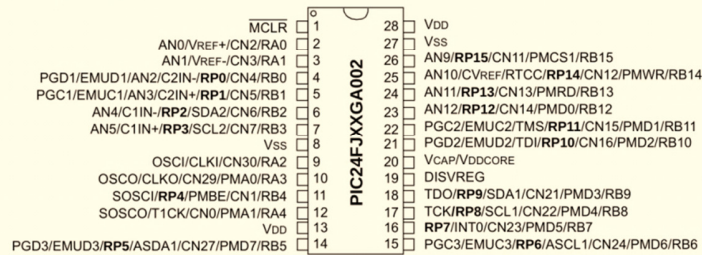
- Peripheral Pin Select:
  - Allows independent I/O mapping of many peripherals
  - Up to 26 available pins (44-pin devices)
  - Continuous hardware integrity checking and safety interlocks prevent unintentional configuration changes
- 8-Bit Parallel Master/Slave Port (PMP/PSP):
  - Up to 16-bit multiplexed addressing, with up to 11 dedicated address pins on 44-pin devices
  - Programmable polarity on control lines
- Hardware Real-Time Clock/Calendar (RTCC):
  - Provides clock, calendar and alarm functions
- Programmable Cyclic Redundancy Check (CRC)
- Two 3-Wire/4-Wire SPI modules (support 4 Frame modes) with 8-Level FIFO Buffer
- Two I<sup>2</sup>C™ modules support Multi-Master/Slave mode and 7-Bit/10-Bit Addressing
- Two UART modules:
  - Supports RS-485, RS-232, and LIN 1.2
  - On-chip hardware encoder/decoder for IrDA®
  - Auto-wake-up on Start bit
  - Auto-Baud Detect
  - 4-level deep FIFO buffer
- Five 16-Bit Timers/Counters with Programmable Prescaler
- Five 16-Bit Capture Inputs
- Five 16-Bit Compare/PWM Outputs
- Configurable Open-Drain Outputs on Digital I/O Pins
- Up to 4 External Interrupt Sources

PIC24FJ Device	Pins	Program Memory (bytes)	SRAM (bytes)	Remappable Peripherals						I <sup>2</sup> C™	10-Bit A/D (ch)	Comparators	PMP/PSP	JTAG
				Remappable Pins	Timers 16-Bit	Capture Input	Compare/PWM Output	UART w/ IrDA®	SPI					
16GA002	28	16K	4K	16	5	5	5	2	2	2	10	2	Y	Y
32GA002	28	32K	8K	16	5	5	5	2	2	2	10	2	Y	Y
48GA002	28	48K	8K	16	5	5	5	2	2	2	10	2	Y	Y
64GA002	28	64K	8K	16	5	5	5	2	2	2	10	2	Y	Y
16GA004	44	16K	4K	26	5	5	5	2	2	2	13	2	Y	Y
32GA004	44	32K	8K	26	5	5	5	2	2	2	13	2	Y	Y
48GA004	44	48K	8K	26	5	5	5	2	2	2	13	2	Y	Y
64GA004	44	64K	8K	26	5	5	5	2	2	2	13	2	Y	Y

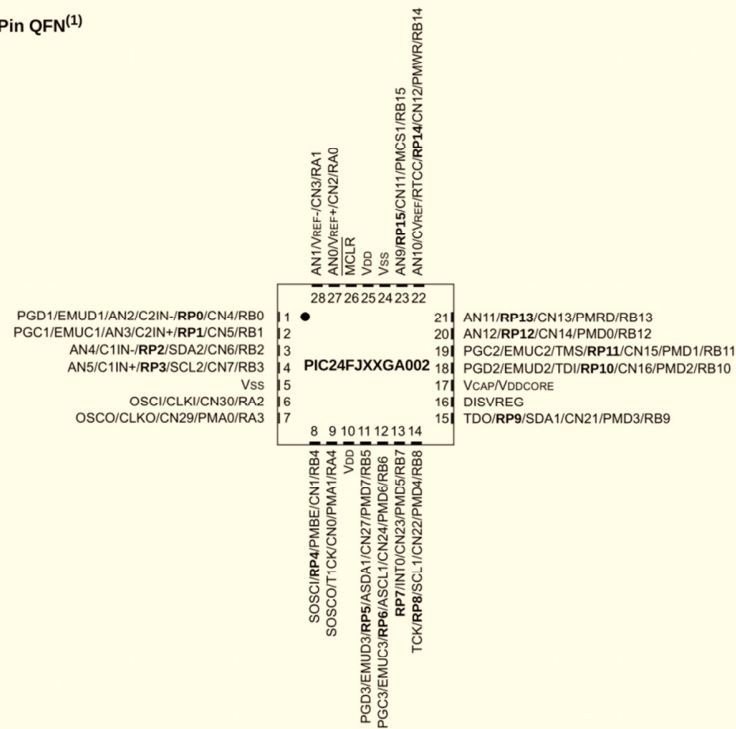
# PIC24FJ64GA004 FAMILY

## Pin Diagrams

### 28-Pin SPDIP, SSOP, SOIC



### 28-Pin QFN<sup>(1)</sup>

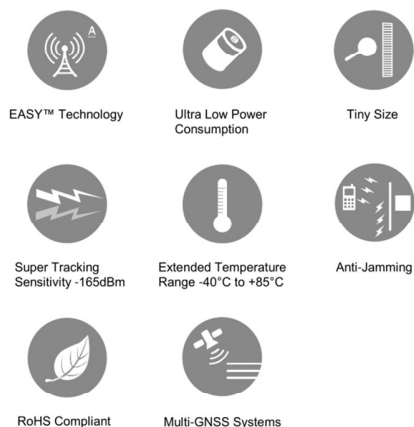


**Legend:** RPN represents remappable peripheral pins.

**Note 1:** Back pad on QFN devices should be connected to Vss.

## Quectel L76

### Compact GNSS Modules



#### Key benefits

- ☞ Extremely compact size, 10.1 × 9.7 × 2.5mm
- ☞ Multi-GNSS engine for GPS, GLONASS, BeiDou, Galileo and QZSS
- ☞ EASY™, advanced AGPS technology without external memory
- ☞ Ultra low tracking power consumption, 18mA
- ☞ Always Locate™, an intelligent algorithm for power saving
- ☞ LOCUS, innate logger solution with no need for host and external flash
- ☞ High sensitivity, -165dBm@Tracking, -148dBm@Acquisition
- ☞ 99 acquisition/33 tracking channels, up to 210PRN channels
- ☞ Support DGPS, SBAS(WAAS/EGNOS/MSAS/GAGAN)
- ☞ Anti-Jamming, Multi-tone Active Interference Canceller
- ☞ Balloon mode, for high altitude up to 80km
- ☞ PPS VS. NMEA can be used in time service

**QUECTEL**®  
Build a Smarter World

GNSS

L76 series, the smallest GNSS module, is concurrent GNSS receivers and can receive and track multiple GNSS systems. With 33 tracking channels, 99 acquisition channels, and 210 PRN channels, L76 series can acquire and track any mix of GNSS signals. L76 series module contains 3 variants: L76(GPS+GLONASS), L76B(GPS+BeiDou) and L76G(GPS+GLONASS+Galileo)\*, which provides enough combination of GNSS to meet the demands of different areas.

Compared with using GPS only, enabling multiple GNSS systems generally increases the number of visible satellites, reduces the time to first fix and increases positioning accuracy, especially when driving in rough urban environments.

Combining advanced AGPS called EASY™ (Embedded Assist System) and proven AlwaysLocate™ technology, L76 series achieves the highest performance and fully meets the industrial standard. Additional feature of embedded logger function called LOCUS allows L76 series to log position information to internal flash memory at default intervals of 15 seconds and provide typically more than 16 hours log capacity without adding cost.

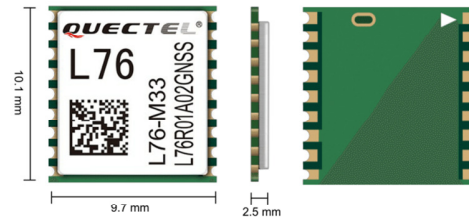
Its super performance makes L76 series ideal for automotive, industrial PDA, consumer and industry applications. Extremely low power consumption makes it easier to be applied to power sensitive devices, especially portable applications.

\* L76G is under development.



## Quectel L76

### Compact GNSS Module



#### General Specifications

<b>L1 Band Receiver</b> (1575.42MHz)	Channel	33 (Tracking) / 99 (Acquisition) Up to 210 (PRN)
	C/A Code	
	SBAS	WAAS, EGNOS MSAS, GAGAN
<b>Horizontal Position Accuracy</b>	Autonomous	<2.5 m CEP
<b>Velocity Accuracy</b>	Without Aid	<0.1m/s
<b>Acceleration Accuracy</b>	Without Aid	0.1m/s <sup>2</sup>
<b>Timing Accuracy</b>	1PPS Out	10ns
<b>Reacquisition Time</b>		<1s
<b>TTFF@-130dBm with EASY™</b>	Cold Start	<15s
	Warm Start	<5s
	Hot Start	<1s
<b>TTFF@-130dBm without EASY™</b>	Cold Start	<35s
	Warm Start	<30s
	Hot Start	<1s
<b>Sensitivity</b>	Acquisition	-148dBm
	Tracking	-165dBm
	Reacquisition	-160dBm
<b>Environmental</b>	Operating Temperature	-40 °C to 85 °C
	Storage Temperature	-45 °C to 125 °C
<b>Dynamic Performance</b>	Maximum Altitude	Max.18000m
	Maximum Velocity	Max.515m/s
	Maximum Acceleration	4G
<b>Dimensions</b>	10.1 x 9.7 x 2.5mm	
<b>Weight</b>	Approx. 0.6g	

#### Power Management

<b>Power Supply</b>	2.8V ~ 4.3V
<b>Power Acquisition</b>	21mA(GPS)
	25mA(GPS+GLONASS)
	23mA(GPS+BeiDou)
<b>Power Tracking</b>	15mA(GPS)
	18mA(GPS+GLONASS)
	18mA(GPS+BeiDou)
<b>Power Saving</b>	Typ.2.6mA @AlwaysLocate™ (Note1)
	7uA@Backup Mode
	500uA@Standby Mode
	Periodic Mode
<b>Antenna Type</b>	Active or Passive
<b>Antenna Power</b>	External or Internal VCC_RF

Note1: Measured in GPS+GLONASS system under outdoor static mode.

#### Serial Interfaces

<b>Serial Interfaces</b>	UART: Adjustable 4800~115200 bps Default: 9600bps
<b>Update rate</b>	1Hz (default), up to10Hz
<b>I/O Voltage</b>	2.7V ~ 2.9V
<b>Protocols</b>	NMEA 0183
	PMTK



Copyright © 2015 Quectel Wireless Solutions Co., Ltd. All Rights Reserved. <http://www.quectel.com>

HQ address: Office 501, Building 13, No.99  
Tianzhou Road, Shanghai, China 200233  
Tel: +86 21 51066236  
Fax: +86 21 54453668  
Email: [info@quectel.com](mailto:info@quectel.com)



Topside View

Underside View

## Dominator

AA.161.301111

## Specification

Part No.	AA.161.301111
Product Name	<b>Dominator</b> Magnet Mounted GPS-GLONASS Antenna
Feature	1575MHz – 1610MHz 1.8-5.5V 3M RG174 SMA(M) IP67 Rated Custom cables and connectors available RoHS Compliant



## 1. Introduction

This antenna is designed for applications which require high positioning accuracy by combining signals from GPS and

GLONASS systems. High gain wide-band patch antenna on a large integral ground delivers maximum performance.

## 2. Specification

### Electrical

<b>Centre Frequency</b>	1574~1610MHz		
<b>Antenna Gain</b>	26 ± 3dBic @ Zenith @ 1575.42MHz 27 ± 3dBic @ Zenith @ 1602MHz		
<b>Axial Ratio</b>	3.0dB max. @ Zenith @ Center Frequency		
<b>Polarization</b>	RHCP		
<b>VSWR</b>	2.0 max.		
<b>Impedance</b>	50Ω		
<b>DC input</b>	1.8V (min.)	3.0V (typ.)	5.5V (max.)
<b>LNA Gain</b>	22dB	28dB	31dB
<b>Noise Figure</b>	2.6dB	2.6dB	2.6dB
<b>Power Consumption</b>	5mA	10mA	10mA

### Mechanical

<b>Antenna Dimensions</b>	65.7 x 49.7 mm
<b>Housing Material</b>	ABS
<b>Cable</b>	3M RG174 (fully customizable)
<b>Connector</b>	SMA(M) (fully customizable)

### Environmental

<b>Operation Temperature</b>	-40°C to 85°C
<b>Storage Temperature</b>	-40°C to 105°C
<b>Relative Humidity</b>	40% to 95%

## 2 SIM900 Overview

Designed for global market, SIM900 is a quad-band GSM/GPRS engine that works on frequencies GSM 850MHz, EGSM 900MHz, DCS 1800MHz and PCS 1900MHz. SIM900 features GPRS multi-slot class 10/ class 8 (optional) and supports the GPRS coding schemes CS-1, CS-2, CS-3 and CS-4.

With a tiny configuration of 24mm x 24mm x 3mm, SIM900 can meet almost all the space requirements in your applications, such as M2M, smart phone, PDA and other mobile devices.

The physical interface to the mobile application is a 68-pin SMT pad, which provides all hardware interfaces between the module and customers' boards.

- The keypad and SPI display interface will give you the flexibility to develop customized applications.
- Serial port and Debug port can help you easily develop your applications.
- One audio channel includes a microphone input and a speaker output.
- Programmable General Purpose Input & Output.

The SIM900 is designed with power saving technique so that the current consumption is as low as 1.0mA in SLEEP mode.

The SIM900 is integrated with the TCP/IP protocol; extended TCP/IP AT commands are developed for customers to use the TCP/IP protocol easily, which is very useful for those data transfer applications.

### 2.1 SIM900 Key Features

Table 3: SIM900 key features

Feature	Implementation
Power supply	Single supply voltage: 3.1V – 4.8V
Power saving	Typical power consumption in SLEEP mode is 1.0mA ( BS-PA-MFRMS=9 )
Frequency Bands	<ul style="list-style-type: none"> <li>● SIM900 Quad-band: GSM 850, EGSM 900, DCS 1800, PCS 1900. The SIM900 can search the 4 frequency bands automatically. The frequency bands also can be set by AT command.</li> <li>● Compliant to GSM Phase 2/2+</li> </ul>
GSM class	Small MS
Transmitting power	<ul style="list-style-type: none"> <li>● Class 4 (2W) at GSM 850 and EGSM 900</li> <li>● Class 1 (1W) at DCS 1800 and PCS 1900</li> </ul>
GPRS connectivity	<ul style="list-style-type: none"> <li>● GPRS multi-slot class 10 ( default )</li> <li>● GPRS multi-slot class 8 (option)</li> <li>● GPRS mobile station class B</li> </ul>

#### SIM900 Hardware Design

Temperature range	<ul style="list-style-type: none"> <li>● Normal operation: -30°C to +80°C</li> <li>● Restricted operation: -40°C to -30°C and +80 °C to +85°C<sup>(1)</sup></li> <li>● Storage temperature -45°C to +90°C</li> </ul>
DATA GPRS:	<ul style="list-style-type: none"> <li>● GPRS data downlink transfer: max. 85.6 kbps</li> <li>● GPRS data uplink transfer: max. 42.8 kbps</li> <li>● Coding scheme: CS-1, CS-2, CS-3 and CS-4</li> <li>● Supports the protocols PAP (Password Authentication Protocol) usually used for PPP connections.</li> <li>● Integrates the TCP/IP protocol.</li> <li>● Support Packet Switched Broadcast Control Channel (PBCCH)</li> </ul>
CSD:	<ul style="list-style-type: none"> <li>● CSD transmission rates: 2.4, 4.8, 9.6, 14.4 kbps, non-transparent</li> <li>● Unstructured Supplementary Services Data (USSD) support</li> </ul>
SMS	<ul style="list-style-type: none"> <li>● MT, MO, CB, Text and PDU mode</li> <li>● SMS storage: SIM card</li> </ul>
FAX	Group 3 Class 1
SIM interface	Support SIM card: 1.8V, 3V
External antenna	Antenna pad
Audio features	Speech codec modes: <ul style="list-style-type: none"> <li>● Half Rate (ETS 06.20)</li> <li>● Full Rate (ETS 06.10)</li> <li>● Enhanced Full Rate (ETS 06.50 / 06.60 / 06.80)</li> <li>● Adaptive multi rate (AMR)</li> <li>● Echo Cancellation</li> <li>● Noise Suppression</li> </ul>
Serial port and Debug port	Serial Port: <ul style="list-style-type: none"> <li>● 8-wire modem interface with status and control lines, unbalanced, asynchronous.</li> <li>● 1.2kbps to 115.2kbps.</li> <li>● Serial Port can be used for AT commands or data stream.</li> <li>● Supports RTS/CTS hardware handshake and software ON/OFF flow control.</li> <li>● Multiplex ability according to GSM 07.10 Multiplexer Protocol.</li> <li>● Autobauding supports baud rate from 1200 bps to 57600bps.</li> </ul> Debug port: <ul style="list-style-type: none"> <li>● 2-wire null modem interface DBG_TXD and DBG_RXD.</li> <li>● Can be used for debugging and upgrading firmware.</li> </ul>
Phonebook management	Support phonebook types: SM, FD, LD, RC, ON, MC.
SIM Application Toolkit	Support SAT class 3, GSM 11.14 Release 99
Real time clock	Implemented
Timer function	Programmable via AT command
Physical characteristics	Size: 24mm x 24mm x 3mm Weight: 3.4g
Firmware upgrade	Firmware upgrade by debug port.

(1) The SIM900 does work, but deviations from the GSM specification may occur.

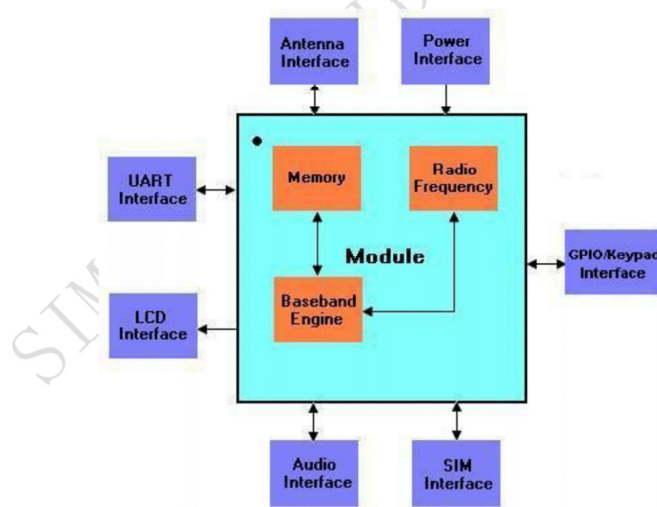
**Table 4: Coding schemes and maximum net data rates over air interface**

Coding scheme	1 Timeslot	2 Timeslot	4 Timeslot
CS-1:	9.05kbps	18.1kbps	36.2kbps
CS-2:	13.4kbps	26.8kbps	53.6kbps
CS-3:	15.6kbps	31.2kbps	62.4kbps
CS-4:	21.4kbps	42.8kbps	85.6kbps

## 2.2 SIM900 Functional Diagram

The following figure shows a functional diagram of the SIM900 and illustrates the mainly functional part:

- The GSM baseband engine
- Flash and SRAM
- The GSM radio frequency part
- The antenna interface
- The Other interfaces



**Figure 1: SIM900 functional diagram**

## Quad-band Wired Cellular Antenna SMA

CEL-00290



**Description:** A Quad-band antenna for embedded cellular devices. This antenna will allow you to connect your cellular module to the outside world. Operates on four frequencies. Comes with a 3M adhesive backing for mounting. Standard male SMA terminated.

### Key Specifications:

- Quad-band: 850/900/1800/1900MHz and 2.4GHz
- For GSM/GPRS, CDMA/1X, TDMA/AMPS, CDPD, 2 Way pager Re-Flex
- Cable with RG174A/U - 3m long
- Connector: SMA



© images are CC BY-NC-SA 3.0

### Electrical Properties:

- Frequency range: 824 ~ 960 MHz // 1710 ~ 1990 MHz
- Impedance: 50 Ohm nominal
- VSWR: <2.0 : 1
- Gain: 3.5dBi
- Radiation: Omni
- Polarization: Vertical
- Wave: Half Wave Dipole

### Mechanical Properties:

- Size: 115x22x4mm
- Connector SMA (Male) standard
- Cable: RG174
- Cable length: 3m
- Over-mold housing: ABS Black
- Mounting: Adhesive backing
- Operation Temperature: -40 to +85C